Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Srovnání virtualizačních technologií

Comparison of Virtualization Technologies

2020 **Bc. Patrik Nechajev**

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání diplomové práce

Student: **Bc. Patrik Nechajev**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2601T013 Telekomunikační technika

Téma: Srovnání virtualizačních technologií
Comparison of Virtualization Technologies

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem diplomové práce je porovnat nástroje pro virtualizaci operačních systémů, srovnat jejich vliv na provoz operačních systémů, způsoby konfigurace a síťové nastavení. Diplomová práce musí obsahovat zpracované následující body zadání:

1. Teoretický rozbor virtualizace operačních systémů. Volba nástrojů pro virtualizaci.
2. Otestováno musí být nejméně 5 nástrojů pro virtualizaci - tzv. "hypervizorů" a nejméně 3 nástroje pro virtualizaci pro operační systémy Windows a Linux.
3. Návrh metodiky pro testování výkonu, využití operační paměti, možnosti využití grafických karet pro výpočty u virtuálních systémů, testování tvorby virtuálních sítí a vliv provozu v těchto sítích na serverovou platformu.
4. Srovnání výkonu virtuálních platforem u výše uvedených parametrů.
5. Analýza nástrojů pro správu virtuálních strojů včetně podrobné dokumentace.
6. Zhodnocení výsledků včetně návrhu optimálního řešení pro virtualizaci.

Seznam doporučené odborné literatury:

[1] PORTNOY, Matthew. *Virtualization essentials*. Second edition. Indianapolis, Indiana: John Wiley, 2016. ISBN 978-1119267720.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Lukáš Kapičák**

Datum zadání: 01.09.2019
Datum odevzdání: 30.04.2020

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA Fakulta elektrotechniky a informatiky 1D16 OSTRAVA

prof. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prehlásenie študenta

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

V Ostrave dňa: *10. května 2020*

.....
podpis študenta

Poďakovanie

Rád by som poďakoval vedúcemu diplomovej práce Ing. Lukášovi Kapičákovi za odbornú pomoc a konzultácie pri vytváraní tejto diplomovej práce.

Ďalej by som sa chcel poďakovať svojej rodine a priateľke za to, že ma počas celej doby môjho štúdia plne podporovali.

Abstrakt

Táto diplomová práca sa zaoberá porovnaním serverových virtualizačných platforiem a platforiem pre operačné systémy Linux a Windows. Cieľom práce je porovnanie vybraných riešení z pohľadu výkonu, možností konfigurácie, administrácie alebo tvorby virtuálnych strojov a sietí. Úlohou teoretickej časti je zoznámiť čitateľa s problematikou virtualizácie a jednotlivých virtualizačných techník. Ďalej nasleduje popis a vlastnosti vybraných serverových virtualizačných platforiem VMware ESXi, Citrix Hypervisor, Microsoft Hyper-V, KVM a LXD. V praktickej časti je popísaný postup inštalácie, tvorby virtuálneho stroja, možnosti a konfigurácia siete a celková príprava pred testovaním. Samotné testovanie zahŕňa podrobný popis nastavnej metodiky, voľbu porovnávacích nástrojov a jednotlivé testy. Po dôkladnom otestovaní serverových platforiem nasleduje stručný popis, vlastnosti, konfigurácia, metodika a jednotlivé testy v menšom rozsahu, ako pri serverovej virtualizácií platforiem pre operačné systémy Linux a Windows, konkrétne VMware Workstation Player, Oracle VM VirtualBox a KVM. V závere sa ešte venujem subjektívnemu porovnaniu nástrojov použitých pre správu virtuálnych strojov a hostiteľa.

Kľúčové slová

VMware; ESXi; KVM; Linux; Microsoft; Hyper-V; Citrix; LXD; VirtualBox; Hypervizor; Virtualizácia; virtuálny stroj; kontajner

Abstract

This diploma thesis deals with comparison of server virtualization platforms and platforms for operating systems Linux and Windows. The main goal of this thesis is comparison of chosen technologies in terms of performance, configuration options, administration or from creation of virtual machines and networks. The teoretical part aim is to inform the reader about virtualization techniques and about virtualization in general. Next this part contains a description of chosen server virtualization technologies such as VMware ESXi, Citrix Hypervisor, Microsoft Hyper-V, KVM and LXD. In the practical part is described process of installation, creation of virtual machine, network capabilities and configuration itself. Testing contain detailed methodology, choice of benchmarks and tests itself. After thorough testing of server virtualization technologies, the practical part continues with virtualization platforms for operating systems Linux and Windows, namely VMware Workstation Player, Oracle VM VirtualBox and KVM. It contains a brief description, features, configuration, methodology and tests itself, but in smaller manner. At the end, i deal with subjective comparison of tools used for managing virtual machines and host.

Key words

VMware; ESXi; KVM; Linux; Microsoft; Hyper-V; Citrix; LXD; VirtualBox; Hypervisor; Virtualization; virtual machine; container

Zoznam použitých skratiek

Skratka	Význam
API	Application Programming Interface
BIOS	Basic Input Output Systém
BTRFS	B-Tree File System
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CAM	Computer-Aided Manufacturing
CIM	Common Information Model
CPU	Central Processing Unit
DCUI	Direct Console User Interface
DHCP	Dynamic Host Configuration Protocol
DMC	Dynamic Memory Control
DNS	Domain Name System
DRS	Distributed Resource Scheduler
FTP	File Transfer Protocol
GPL	General Public License
GPU	Graphical Processing Unit
GVT	Graphics Virtualization Technology
HDD	Hard Disk Drive
HTML 5	HyperText Markup Language 5
IaaS	Infrastructure as a Service
ICMP	Internet Control Message Protocol
KVM	Kernel-based Virtual Machine
LVM	Logical Volume Manager
LXC	Linux Containers
NAT	Network Address Translation
NFS	Network File System
NIC	Network Interface Card
NTP	Network Time Protocol
OS	Operating System

PIF	Physical Network Interface
POSIX	Portable Operating System Interface
QEMU	Quick Emulator
RAM	Random Access Memory
RDP	Remote Desktop Protocol
REST	Representational State Transfer
RST	Remote Server Administration Tools
SCP	Secure Copy
SCTP	Stream Control Transmission Protocol
SDN	Software-Defined Networking
SMB	Server Message Block
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
SR-IOV	Single Root Input/Output Virtualization
SSD	Solid State Drive
SSH	Secure Shell
SVM	Secure Virtual Machine
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UEFI	Unified Extensible Firmware Interface
UID	User Identifier
URL	Uniform Resource Locator
VCPU	Virtual Central Processing Unit
VDI	Virtual Desktop Infrastructure
VGPU	Virtual Graphical Processing Unit
VIF	Virtual Network Interface
VM	Virtual Machine
VMFS6	Virtual Machine File System 6
VMM	Virtual Machine Monitor
VNC	Virtual Network Computing
XML	eXtensible Markup Language

Zoznam obrázkov

Zoznam tabuliek

Zoznam grafov

Obsah

Úvod

Serverová virtualizácia je v oblasti informačných technologií veľmi rozšíreným pojmom. S neustále narastajúcim výkonom serverov spočíva jej účel hlavne v maximálnom a efektívnom využití dostupných zdrojov. Mimo iného hovoríme aj o zjednodušenej administrácií, znížení prevádzkových nákladov a zvýšení bezpečnosti.

Diplomová práca je zameraná na porovnanie piatich súčasných serverových virtualizačných platforiem a na porovnanie troch platforiem pre operačné systémy Microsoft Windows a Linux z pohľadu výkonu, možností konfigurácie alebo spôsobu administrácie.

Prvá kapitola obsahuje teoretický popis serverovej virtualizácie, popis typov rôznych virtualizačných techník, ako plná virtualizácia, paravirtualizácia a virtualizácia na úrovni operačného systému. Ďalej je v tejto kapitole objasnený pojem Hypervizor a jeho typy.

Druhá kapitola obsahuje rešerš vybraných serverových virtualizačných platforiem. Konkrétne sa jedná o platformy VMware ESXi, Citrix Hypervisor, Microsoft Hyper-V, KVM a LXD. Pri každej z platforiem nájdeme popis, vlastnosti, architektúru, prípadne stručnú históriu. V poslednej rade obsahuje táto kapitola aj popis jednotlivých nástrojov pre správu virtuálnych strojov.

Tretia kapitola obsahuje teoretický popis možností využitia grafických kariet pre virtualizáciu. Zahrnutý je popis, rozdelenie, možnosti, prípadne architektúra technológii NVIDIA vGPU, AMD MxGPU a Intel GVT.

Štvrtá kapitola obsahuje špecifikáciu poskytnutých a použitých hardvérových prostriedkov, tabuľku s prehľadom pridelených IP adries a následné rozdelenie serverových virtualizačných platforiem vrátane konkrétnej verzie. Taktiež sa tu nachádza voľba operačného systému virtuálnych strojov.

Piata kapitola obsahuje inštaláciu a konfiguráciu jednotlivých virtualizačných platforiem. Úvod kapitoly je zameraný na podrobný popis tvorby brány do internetu založenej na Ubuntu Server 18.04.3 LTS pre virtuálne stroje. Nasleduje stručný popis procesu inštalácie zvolených virtualizačných platforiem. Kapitola ďalej obsahuje proces tvorby virtuálneho stroja, možnosti a konfiguráciu siete. Celý proces inštalácie, tvorby a konfigurácie je doplnený o názorné ukážky a príkazy.

Šiesta kapitola obsahuje voľbu virtualizačných platforiem pre operačné systémy Windows a Linux, konkrétne sa jedná o VMware Workstation Player, Oracle VM VirtualBox a KVM. Podobne, ako štvrtá kapitola obsahuje špecifikáciu hardvérových prostriedkov a voľbu operačného systému virtuálnych strojov. Pri týchto platformách bol použitý môj súkromný prenosný počítač.

Siedma kapitola obsahuje štručný popis, inštaláciu a konfiguráciu virtualizačných platforiem pre OS Windows a Linux.

Ôsma kapitola obsahuje návrh metodiky testovania a výkonnostné testy serverových virtualizačných platforiem. Zahrnuté sú testy CPU, operačnej pamäte, súborového systému a virtuálnej siete. Medzi zvolenými porovnávacími nástrojmi sa nachádzajú napríklad Sysbench, RAMspeed SMP, IOzone alebo iPerf3. Každý z testov obsahuje podrobnú metodiku testovania, ktorá bola pri danom teste nastavená. Jednotlivé testy sú vždy okomentované a podkapitoly ukončené stručným zhrnutím.

Mimo výkonnostných testov obsahuje ôsma kapitola aj využitie a možnosti správy operačnej pamäte a je zameraná aj na vplyv prevádzky vo virtuálnej sieti na serverovú platformu.

Deviata kapitola obsahuje návrh metodiky a výkonnostné testy virtualizačných platforiem pre operačné systémy Windows a Linux. Pre platformy VMware Workstation Player a Oracle VM VirtualBox boli zvolené dva hostiteľské systémy, Windows 10 a Ubuntu 19.10. Týmto sa naskytla možnosť sledovať nie len rozdiely medzi platformami, ale aj vplyv hostiteľského OS na výsledky rovnakej platformy. Podobne, ako pri serverových riešeniach je každý test okomentovaný a každá podkapitola ukončená stručným zhrnutím.

Desiata kapitola obsahuje subjektívne porovnanie nástrojov použitých ku správe virtuálnych strojov a hostiteľa. Zahŕňa výhody, nevýhody a celkový dojem z používania počas doby, ktorej som s nástrojmi pracoval.

Záver obsahuje zhodnotenie dosiahnutých výsledkov a poznatkov, získaných počas testovania jednotlivých virtualizačných platforiem.

1 Úvod do virtualizácie

Čo si vlastne pod pojmom virtualizácia predstaviť? Zjednodušene povedané nám virtualizácia umožňuje súbežný beh viacerých izolovaných virtuálnych strojov nad jedným fyzickým strojom. Tieto virtuálne stroje sa spolu delia o hardvérové prostriedky, ako je procesor, pamäť, harddisky alebo sieť. To, že sú jednotlivé virtuálne stroje izolované nám umožní beh rôznych OS, ako Linux a Windows, taktiež pád jedného OS neovplyvní beh ostatných OS[1].

Takže na jednom fyzickom stroji mám viacero virtuálnych strojov, k čomu je to teda dobré? Ako hlavné výhody virtualizácie môžeme uviesť[2]:

- Efektívnosť využitia HW.
- Nižšie náklady na údržbu HW.
- Vyššia bezpečnosť.
- Nižšie náklady na spotrebu eletrickej energie.
- Jednoduchší prehľad a správa strojov.
- Jednoduchšia záloha a obnova serverov.
- Rýchlejšie nasadzovanie serverov do produkcie.

1.1 Pojem Hypervizor

Na fyzický hardvér sa nainštaluje takzvaný Hypervizor, môžeme ho definovať ako softvérovú vrstvu, ktorej hlavnou úlohou je prideľovanie prostriedkov jednotlivým virtuálnym strojom, ktorým poskytuje virtuálny hardvér. Administrátor cez hypervizora týmto virtuálnym strojom pridelí určitý počet CPU/jadier, virtuálny HDD, pamäť RAM a podobne.

Existujú dva typy hypervizorov a to Typ 1 a Typ 2. V tejto práci som sa stretol s oboma riešeniami.

1.1.1 Hypervizor - Typ 1

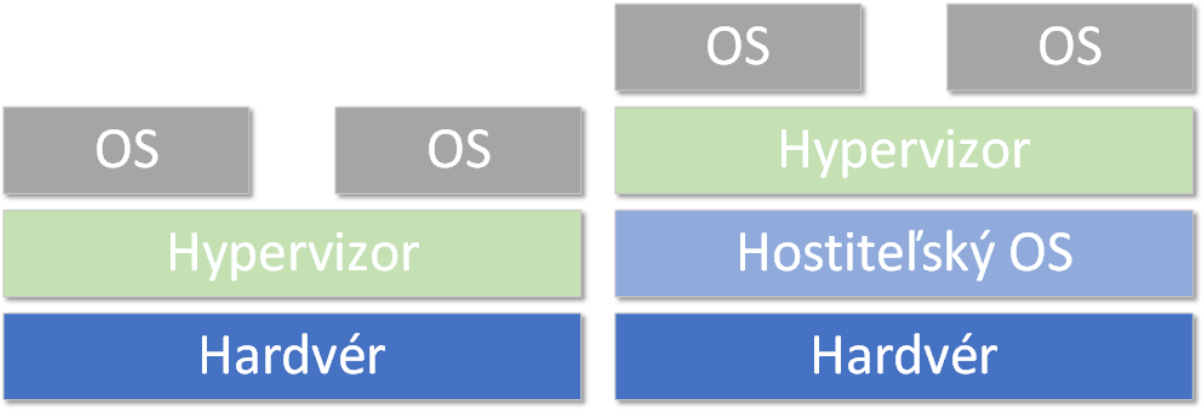
Hypervizor Typu 1 sa inštaluje priamo na fyzický hardvér. Z toho vznikol aj jeho názov „bare metal", nakoľko ku svojej činnosti nutne nepotrebuje žiaden základný hostiteľský systém pod ním[3]. Vďaka tomu, že má priamy prístup k fyzickému hardvéru je menej náchylný na bezpečnostné chyby a rôzne zraniteľnosti. Je považovaný za najefektívnejšie a najvýkonnejšie riešenie pre podnikovú sféru.

Hypervizor Typu 1 využívajú napríklad: Citrix Hypervisor, Microsoft Hyper-V alebo VMware ESXi.

1.1.2 Hypervizor - Typ 2

Hypervizor Typu 2 sa inštaluje ako softvérová aplikácia na už existujúci základný hostiteľský OS. Z toho vznikol aj jeho názov „hosted hypervisor"[3]. Hypervizor Typu 2 závisí na hostiteľskom OS z pohľadu spravovania CPU, úložného priestoru, pamäte a siete. Hypervizor Typu 2 nie je vhodný pre podnikovú sféru, ale používa sa hlavne ako testovacie prostredie v domácej sfére, kde nie sú kladené tak vysoké požiadavky na bezpečnosť a výkon, ako v prípade hypervizora Typu 1.

Hypervizor Typu 2 využívajú napríklad: VMware Workstation Player alebo Oracle VM VirtualBox.



Obrázok 1: Porovnanie jednotlivých typov hypervizorov, Typ 1(vľavo) a Typ 2(vpravo)

1.2 Typy serverovej virtualizácie

Serverová virtualizácia nám umožňuje rozdeliť jeden fyzický server do viacero menších virtuálych serverov, z ktorých každý používa svoj vlastný operačný systém[4]. Tieto operačné systémy sú známe ako hosťované operačné systémy.

Serverovú virtualizáciu môžeme rozdeliť na 3 základné skupiny:

- Plná virtualizácia, tá sa delí na softvérovú a hardvérovú virtualizáciu.
- Paravirtualizácia.
- Virtualizácia na úrovni operačného systému (Kontajnerová virtualizácia).

1.2.1 Plná virtualizácia

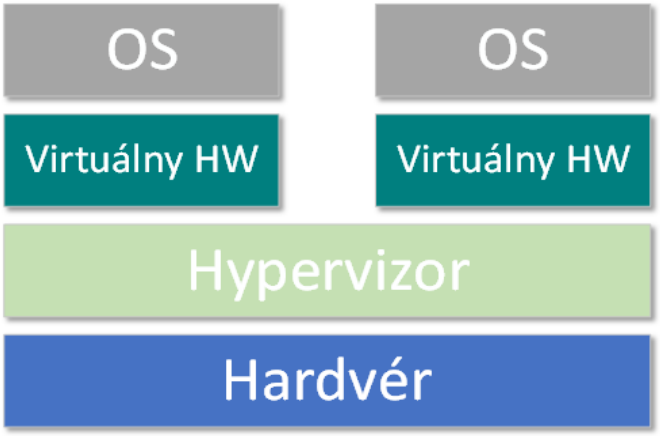
Pre každý hosťovaný systém sa vytvára identický obraz fyzickej architektúry. V takomto prípade má hosťovaný operačný systém k dispozicií rovnakú inštrukčnú sadu a prostriedky, ako na fyzickom stroji a nemôže spoznať, že beží vo virtuálnom prostredí a zdieľa prístup k hardvéru s ďalšími strojmi. Operačný systém, a ani aplikácie nepotrebujú žiadne modifikácie. Môžeme povedať, že sa v podstate jedná o ideálny stav, kedy dochádza k plnému oddeleniu fyzickej vrstvy[5].

Príklady: VMware ESXi, KVM, Microsoft Hyper-V, Citrix Hypervisor,VMware Workstation Player, Oracle VM VirtualBox.

1.2.1.1 Hardvérová virtualizácia

Hardvérová virtualizácia eliminuje binárny preklad a priamo komunikuje s hardvérom vďaka použitiu virtualizačných rozšírení Intel VT alebo AMD-V, ktoré sú priamo integrované v procesore[6]. Virtuálne stroje bežia priamo nad jedným fyzickým hardvérom bez nutnosti už existujúceho základného hostiteľského operačného systému. Prerozdelenie a sprístupnenie hardvérových prostriedkov má na starosti hypervizor.

Príklady: VMware ESXi, KVM, Microsoft Hyper-V, Citrix Hypervisor, VMware Workstation Player (64-bit hostia), Oracle VM VirtualBox (64-bit hostia).

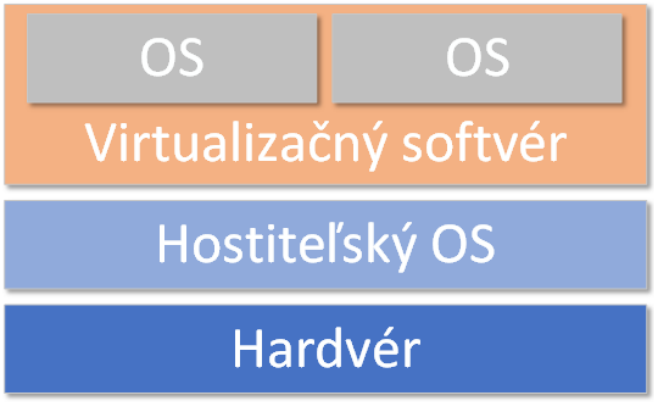


Obrázok 2: Hardvérová virtualizácia

1.2.1.2 ***Softvérová virtualizácia***

Tento typ virtualizácie kompletne závisí na binárnom preklade a emuluje hardvér použitím softvérových inštrukčných sád. Tým je toto riešenie ďaleko menej výkonné v porovnaní s hardvérovou virtualizáciou[6]. Mimo iného, softvérová virtualizácia využíva už existujúci základný operačný systém na ktorom je nainštalovaný virtualizačný softvér v ktorom spúšťa jednotlivé virtuálne stroje.

Príklady: Oracle VM VirtualBox (32-bit hostia), VMware Workstation Player (32-bit hostia).

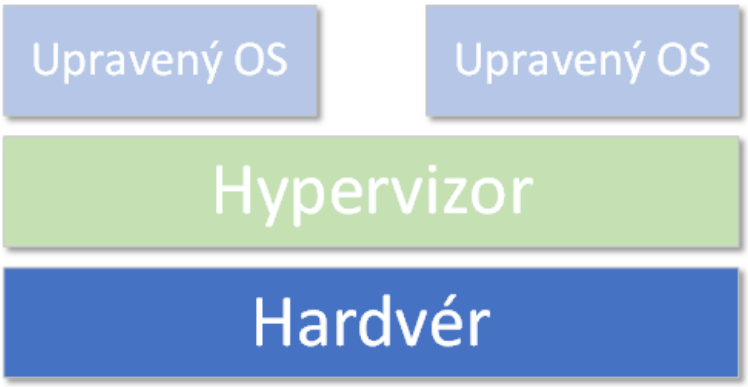


Obrázok 3: Softvérová virtualizácia

1.2.2 **Paravirtualizácia**

Paravirtualizácia sa vyznačuje tým, že vykonáva len čiastočnú abstrakciu na úrovni virtuálneho počítača. Môžeme povedať, že ponúka prostredie, ktoré je podobné tomu fyzickému, na ktorom virtuálny počítač prevádzkujeme. Paravirtualizácia teda nie je úplná, niektoré vlastnosti, napríklad procesoru môžu byť obmedzené. Narozdiel od plnej virtualizácie, operačný systém rozpozná, že beží vo virtuálnom prostredí a dokáže efektívne komunikovať s hypervizorom[6]. Základom pre túto efektívnu komunikáciu je ale špeciálne upravené jadro hosťovaného systému. Práve táto vlastnosť para-virtualizácie komplikuje jej nasadenie pri operačných systémoch s uzavretým zdrojovým kódom. Čiastočne sa ale dá dosiahnuť v prípade použitia špeciálnych ovládačov.

Príklady: Citrix Hypervisor.

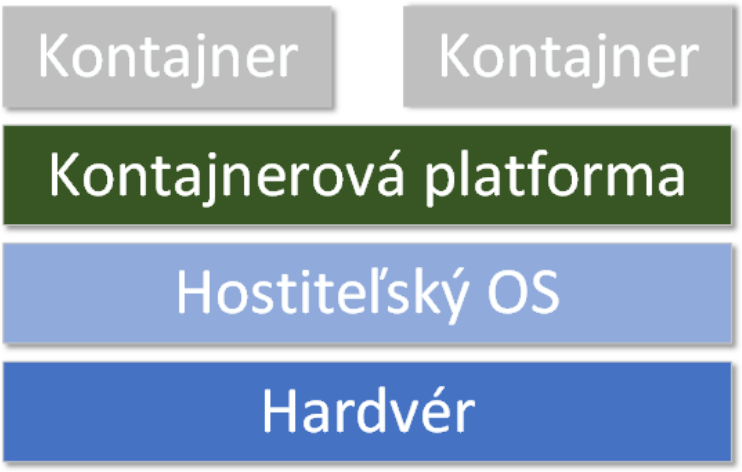


Obrázok 4: Paravirtualizácia

1.2.3 Virtualizácia na úrovni operačného systému

Virtualizácia na úrovni operačného systému, známa aj ako kontajnerová virtualizácia je v dnešnej dobe veľmi rozšírená. V prípade tohto typu virtualizácie nám jadro hostiteľského operačného systému umožňuje beh viacero izolovaných inštancií, tieto inštancie sa inak nazývajú aj kontajnery. Inak povedané sa v rámci jedného operačného systému vytvárajú vzájomne oddelené prostredia, ktoré nám napríklad umožňujú prevádzkovať niekoľko webových serverov bez toho, aby bolo nutné mať pre každý z nich nainštalovaný kompletný systém[7]. Výhodou tohto typu virtualizácie je jej menšia náročnosť na systémové prostriedky, nevýhodou zasa to, že oddelenie nie je úplné, nakoľko kontajnery používajú rovnaký hostiteľský operačný systém a jeho jadro.

Príklady: LXC/LXD, Docker.



Obrázok 5: Virtualizácia na úrovni operačného systému

2 Virtualizačné platformy

Kapitola je zameraná na teoretický popis serverových virtualizačných platforiem. Obsahuje vlastnosti, architektúru, prípadne stručnú históriu. Ďalej sa tu nachádza popis jednotlivých nástrojov určených ku správe infraštruktúry.

2.1 VMware

Je líder v oblasti virtualizácie a cloud komputingu so sídlom v Pao Alto, Kalifornia. VMware bol založený v roku 1998 a je dcérskou spoločnosťou DELL Technologies. Zakľadatelia VMware Diane Greene, Scott Devine, Mendel Rosenblaum, Edward Wang a Edouard Bugnion uviedli v roku 1999 prvý produkt VMware Workstation, nasledovaný VMware ESX v roku 2001, na ktorom VMware dodnes zakladá svoje virtualizačné technológie[8]. VMware mimo virtualizácie ponúka radu rôznych produktov na správu sietí, bezpečnosti alebo data centier.

Hlavným produktom spoločnosti je balík VMware vSphere v aktuálnej verzii 6.7, ktorý obsahuje nasledujúce produkty:

- VMware ESXi - Hypervizor Typu 1.
- VMware vCenter Server - Softvérová aplikácia používaná k správe vSphere infraštruktúry.
- VMware vSphere Client - Používateľské rozhranie slúžiace k vzdialenému pripojeniu k vCenter Server alebo k ESXi.
- VMware vSphere Web Client - Webové používateľske rozhranie slúžiace k vzdialenému pripojeniu k vCenter Server alebo k ESXi z webových prehliadačov a operačných systémov.
- VMware vMotion - Umožňuje prenos už spustených virtuálnych strojov z jedného fyzického serveru na druhý bez nutnosti ich pozastavenia.

Balík VMware vSphere obsahuje tri platené verzie, konkrétne sa jedná o verzie:

- Standard.
- Enterprise Plus.
- Platinum.

Tieto verzie sa odlišujú funkcionalitou a každá ponúka pred kúpou 60-dňovú skúšobnú dobu[9]. VMware ale ponúka aj samostatný hypervizor ESXi, ktorý je dostupný zdarma a ponúka vytvorenie neobmedzeného množstva virtuálnych strojov[10]. To, že je zdarma samozrejme prináša určité obmedzenia, medzi hlavné patrí:

- Žiadna oficiálna podpora zo strany spoločnosti VMware.
- Maximálne 8 vCPU pre jeden virtuálny stroj.
- Nie je možné spravovať pomocou vCenter.

Pri balíku vSphere sme hovorili o čisto serverovom riešení, ale VMware ponúka aj veľmi obľubenú alternatívu pre klasické desktopové počítače, ktorá je ideálna pre domáce testovanie. Jedná sa o VMware Workstation v aktuálnej verzii 15.5.1. Tu narozdiel od ESXi hovoríme o hypervizorovi Typu 2. VMware ponúka dve verzie, neplatenú Workstation Player a platenú Workstation Pro, ktoré sa samozrejme líšia funkcionalitou. VMware Workstation je dostupná pre operačné systémy Windows a Linux. Pre operačný systém macOS existuje samostatná aplikácia VMware Fusion.

2.1.1 Architektúra VMware ESXi

VMware ESXi je „bare metal" hypervizor Typu 1, ktorý sa inštaluje priamo na fyzický hardvér serveru. Vďaka tomu je VMware ESXi schopný efektívne prerozdelovať prostriedky potrebné pre chod jednotlivých virtuálnych strojov. Architektúra VMware ESXi zahŕňa základný operačný systém, nazývaný VMkernel a procesy bežiace nad ním.

VMkernel je operačný systém typu POSIX vyvíjaný spoločnosťou VMware[11]. Poskytuje podobnú funkcionalitu, ktorú poznáme z iných operačných systémov, ako vytváranie procesov a ich kontrola, signalizácia, súborový systém. Je špeciálne navrhnutý pre účely zabezpečenia behu viacerých virtuálnych strojov. VMkernel má kontrolu nad celým fyzickým hardvérom serveru. Hlavné procesy, ktoré bežia nad VMkernelom sú:

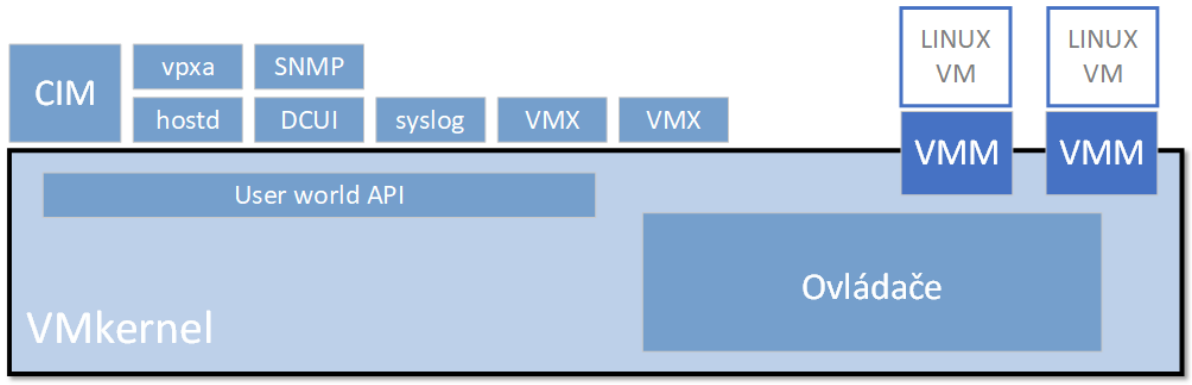
- Direct Console User Interface (DCUI) - je lokálne používateľské rozhranie prístupné cez konzolu ESXi systému, používa sa hlavne pre prvotnú a základnú konfiguáciu serveru. DCUI poskytuje napríklad nastavenie administratívneho hesla, konfiguráciu siete alebo zobrazenie logov.
- Virtual Machine Monitor (VMM) - je proces, ktorý v spolupráci s procesom VMX poskytuje prostredie pre jednotlivé virtuálne stroje. Každý virtuálny stroj má svoj vlastný VMM a VMX proces.
- The Common Information Model (CIM) - CIM je rozhranie, ktoré prostredníctvom API umožňuje vzdialenú správu hardvérových prostriedkov.

User Worlds, termín „user world" odkazuje na proces bežiaci vo VMkernel operačnom systéme. Je to proces, ktorý poskytuje nevyhnutné prostriedky spojené s behom procesov v prostredí hypervizora.

Other User World Processes procesy, ktoré sú riadené VMkernelom. Patria sem:

- Hostd - proces využívaný k overeniu používateľov. Taktiež dohliada na prístupové práva jednotlivých používateľov.
- Vpxa - proces, ktorý zabezpečuje pripojenie k vCenter aplikácii.
- Syslog - zabezpečuje lokálne logovanie informácií s možnosťou odosielania logov na vzdialený server.

ESXi naviac poskytuje NTP synchronizáciu času a monitorovanie pomocou SNMP[11].



Obrázok 6: Architektúra VMware ESXi

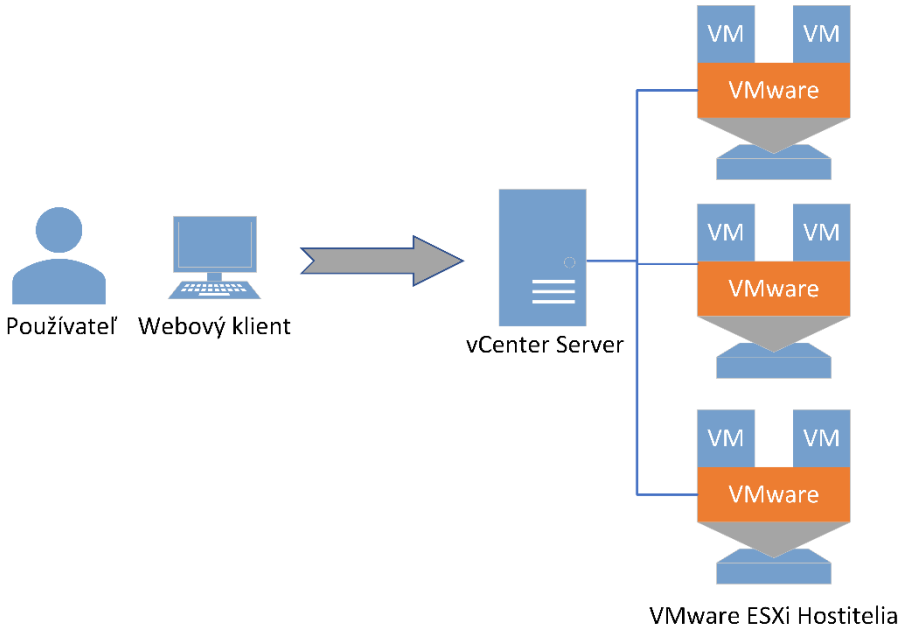
2.1.2 Nástroje pre správu VMware ESXi virtuálnych strojov

VMware vCenter Server je centralizovaná monitorovacia a riadiaca platforma určená pre VMware vSphere infraštruktúru. VMware vCenter Server vykonáva množstvo úloh vrátane poskytovania a prideľovania prostriedkov, monitorovania výkonu, automatizácie, správy prístupových práv používateľov. Umožňuje administrátorovi vSphere infraštruktúry spravovať viacero ESXi serverov, virtuálnych strojov alebo virtuálnych sietí. Niektoré funkcie, ktoré VMware vCenter poskytuje sú:

- vCenter High Availability - Chráni vCenter Server pred zlyhaním hostiteľa a zlyhaniami hardvéru.
- vSphere Distributed Resource Scheduler (DRS) - Vyvažuje zaťaženie serveru s dostupnými zdrojmi, tak aby bol výkon optimálny.

vCenter architektúra pozostáva z dvoch hlavných komponentov a to vCenter Server, ktorý obsahuje služby ako vSphere Web Client alebo vSphere Client. Druhým komponentom je Platform Services Controller, ktorý pozostáva z vCenter Single Sign-On služby poskytujúcej overenie používateľa, ďaľej služby spojené s licencovaním a VMware vSphere certifikačnú autoritu.

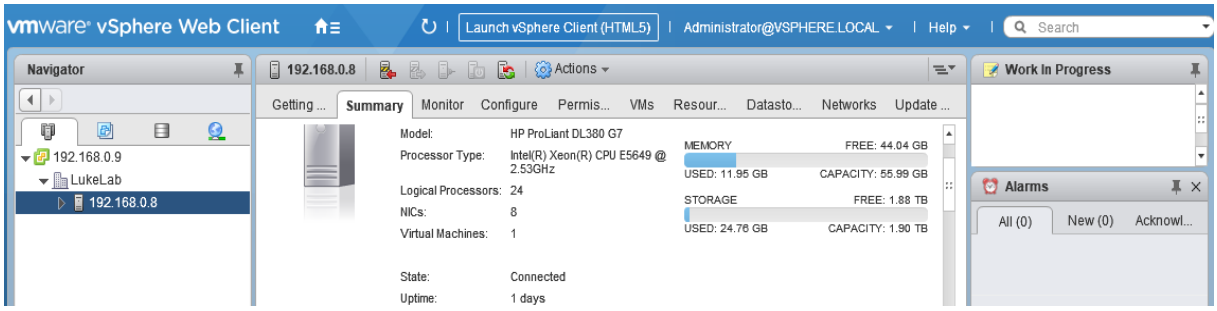
VMware vCenter Server je dostupný v troch platených verziách. Jedná sa o verzie Essentials, ktorá umožňuje správu maximáne 3 hostiteľov, Foundation umožňuje správu maximálne 4 hostiteľov a Standard, ktorá umožňuje neobmedzené množstvo hostiteľov, mimo iného sa samozrejme líšia vo funkciách, ktoré ponúkajú. Administrátor sa môže pripojiť na vCenter server pomocou vSphere Clienta alebo vSphere Web Clienta. V predchádzajúcich vydaniach balíku vSphere sa používal hlavne thick klient, ktorý bol postavený na programovacom jazyku C#. Vývoj tohoto klienta bol od verzie 6.0 ukončený a vo verzii 6.5 nahradený novým, moderným webovým klientom postaveným na HTML 5[12].



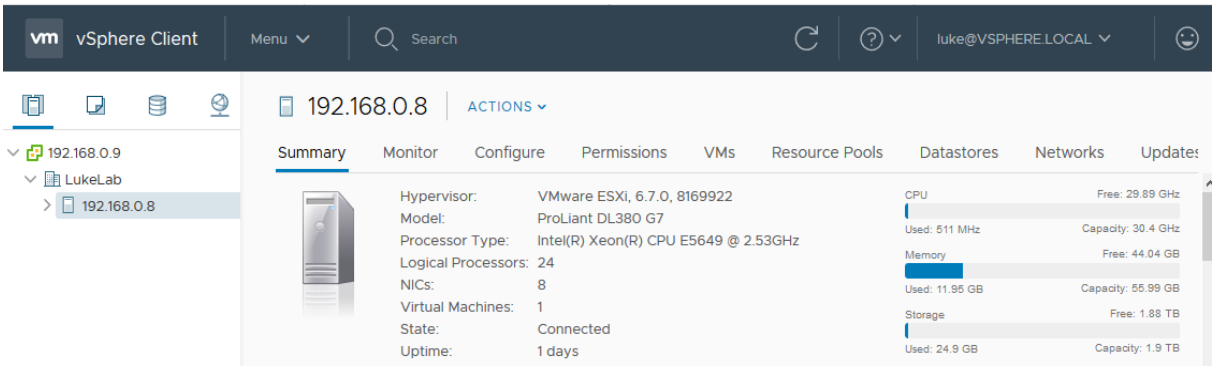
Obrázok 7: vCenter administračná architektúra

vSphere Web Client je starší webový klient založený na Adobe Flex, ktorý ku svojej činnosti potrebuje mať nainštalovaný Adobe Flash[13]. Pre prístup do vSphere Web Clienta zadáme do webového prehliadača URL *https://vcenter1/ui* (ukážkový odkaz).

vSphere Client je moderný webový klient založený na HTML 5, tu odpadá nutnosť inštalácie aplikácie tretej strany Adobe Flash, čo je nesporná výhoda. Jediné, čo tento klient vyžaduje je aktualizovaný webový prehliadač. Pre prístup do vSphere Clienta zadáme do webového prehliadača URL *https://vcenter1/vsphere-client* (ukážkový odkaz).

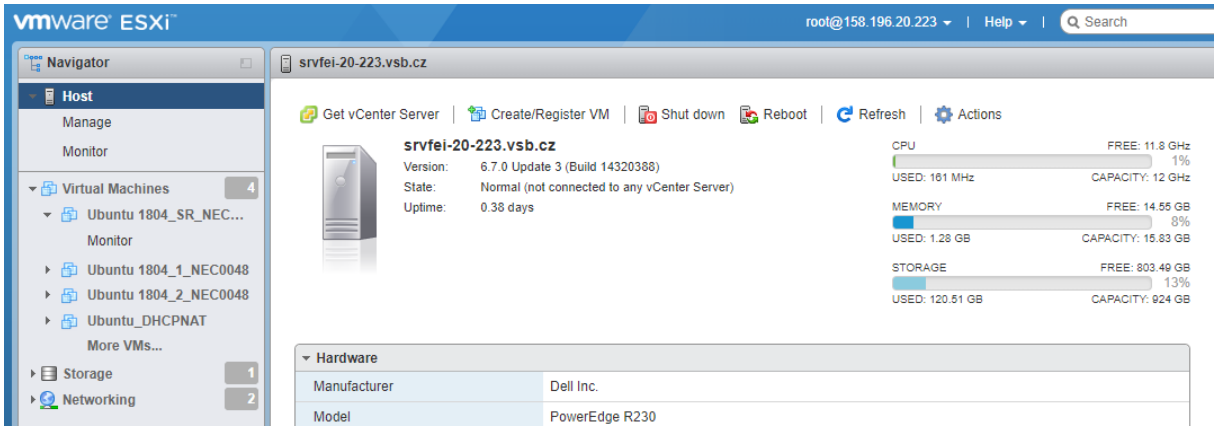


Obrázok 8: Používateľské rozhranie vSphere Web Client[13]



Obrázok 9: Používateľské rozhranie vSphere Client[13]

Nástroje vSphere Web Client a vSphere Client sú nástroje určené len na správu vCenter Servera, ktorý ako už bolo spomenuté nie je zdarma. V prípade, že používame samostatný hypervizor ESXi bez vCenter Serveru, tak musíme použiť ku správe iný nástroj. Jedná sa o nástroj **ESXi Embedded Host Client**. Názov „embedded" napovedá, že sa jedná o vstavaný klient, ktorý je súčasťou hypervizora ESXi. Tento klient je postavený na HTML a JavaScripte, je určený len pre hypervizor ESXi a nie je možné použiť ho ku správe vCenter Serveru[14]. Podporuje všetky moderné aktualizované webové prehliadače a pripojíme sa k nemu podobne ako k vSphere Web Client pomocou URL *https://esxi/ui* (ukážkový odkaz).



Obrázok 10: Používateľské rozhranie ESXi Embedded Client

2.2 Citrix Hypervisor

Citrix Systems je softvérová spoločnosť založená v roku 1989 Edom Iacobuccim. Citrix ponúka rôzne softverové riešenia týkajúce sa serverov, virtualizácie a sietí. V roku 2018 prešla spoločnosť rebrandingom[15]. Príkladom je najznámejší produkt a to virtualizačná platforma XenServer, ktorá bola premenovaná a dnes sa už označuje ako Citrix Hypervisor. Citrix Hypervisor v aktuálnej verzii 8.1 je podobne ako VMware ESXi „bare metal" hypervisor Typu 1. Citrix Hypervisor 8.1 je dostupný v 3 edíciach:

- Premium Edition (pred rebrandingom známa ako Enterprise Edition).
- Standard Edition.
- Express Edition (pred rebrandingom známa ako Free Edition).

Verzia Standard Edition je základné komerčné riešenie, ktoré Citrix ponúka. Ponúka širokú škálu funkcií pre zákazníkov, ktorí chcú robustnú a vysoko výkonnú virtualizačnú platformu, ale nepotrebujú funckie vyššej Premium Edition. Vo verzií Standard Edition je zahrnutá komplexná Citrix podpora a funkcie ako integrácia Active Directory, dynamická kontrola a prideľovanie pamäte virtuálnym strojom alebo ochrana pred zlyhaním hostiteľa.

Verzia Premium Edition je riešenie, optimalizované pre servery a veľké pracovné zaťaženie. Mimo funkcií, ktoré obsahuje Standard Edition, Premium Edition ponúka naviac napríklad automatické aktualizácie ovládačov Windows virtuálnych strojov alebo programy na uľahčenie prechodu z VMware vSphere na Citrix Hypervisor infraštruktúru.

Verzia Express Edition je zdarma a ponúka obmedzené množstvo funkcií bez možnosti Citrix podpory. Express Edition nevyžaduje žiadnu licenciu. Virtuálne stroje su pri využití Express Edition označené v aplikácii XenCenter ako „Unlicensed"[16].

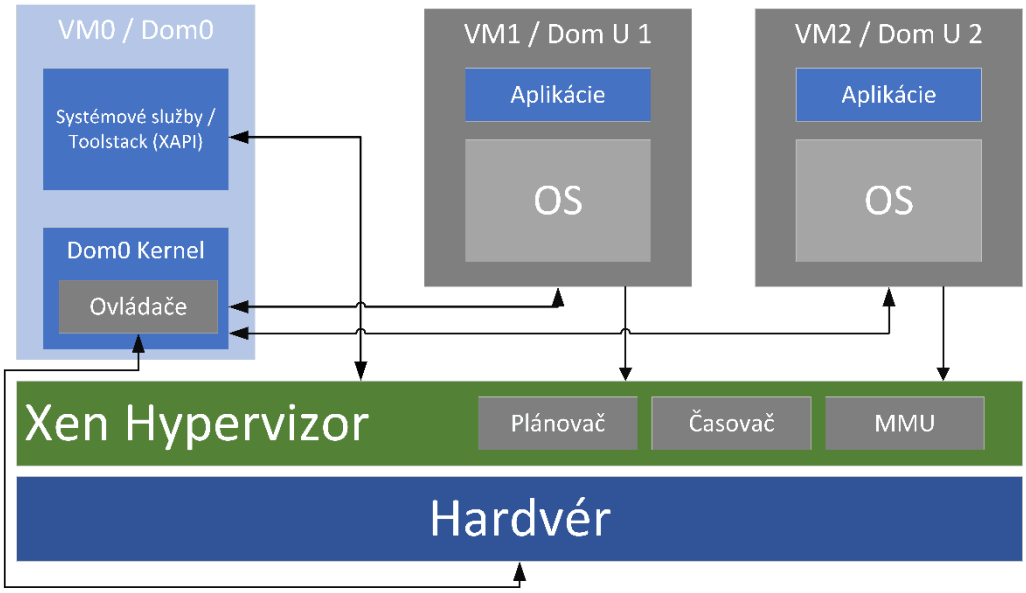
2.2.1 Architektúra Citrix Hypervisor

Xen Project je open-source „bare metal" hypervizor Typu 1, ktorý mimo plnej virtualizácie podporuje aj paravirtualizáciu. Je používaný ako základ pre množstvo komerčných a open-source aplikácií, ako serverová virtualizácia alebo Infraštruktúra ako Služba (IaaS). Citrix Hypervisor je založený na Xen Project hypervizorovi, naviac pridáva služby a podporu zo strany spoločnosti Citrix. Citrix Hypervisor 8.1 je založený na verzii Xen Project hypervisor 4.13.

Control Domain, inak aj **Domain 0** alebo **Dom0** je bezpečný, privilegovaný virtuálny stroj založený na operačnom systéme Linux, konkrétne sa hovoríme o distribúcií CentOS 7.5[17]. Jeho úlohou je zabezpečiť správu a beh XAPI / Toolstacku. Dom0 má priamy prístup k fyzickému hardvéru serveru a zabezpečuje komunikáciu s hypervizorom.

Toolstack, inak aj **XAPI** kontroluje životný cyklus jednotlivých virtuálnych strojov, ďalej kontroluje virtuálne siete, virtuálne úložiská a zabezpečuje overenie používateľa.

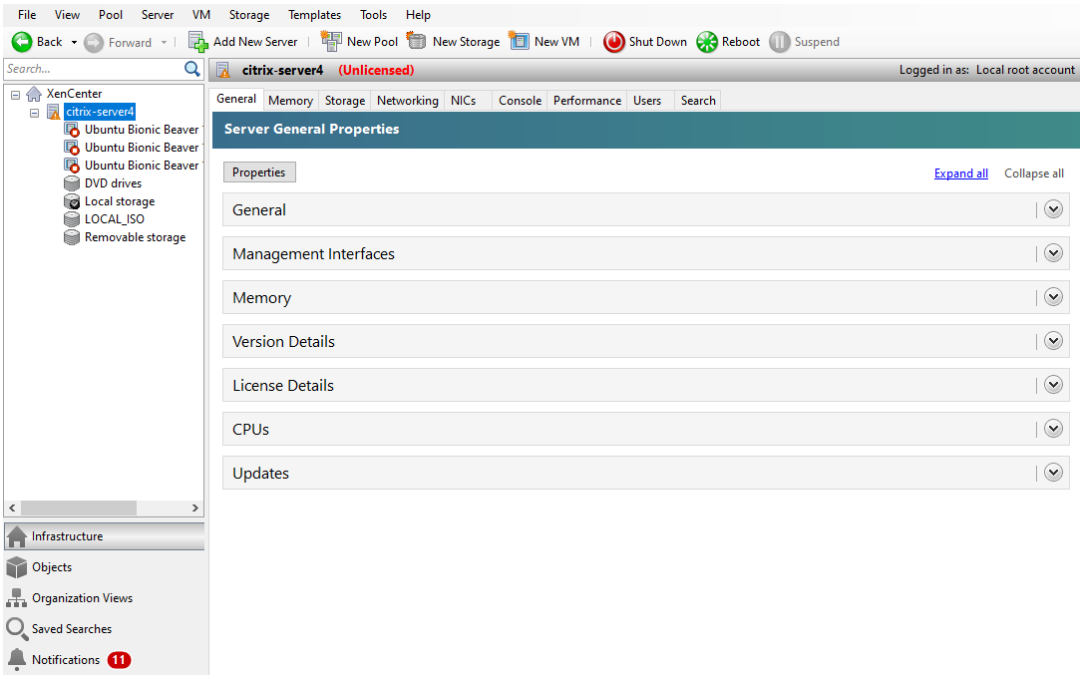
Guest domain (VM) jednotlivé virtuálne stroje sa v prípade Xen Project hypervizora označujú ako domény. Hlavnou doménou je už spomenutá Dom0, ktorá týmto hosťovaným doménam prideľuje prostriedky. Tieto hosťované virtuálne stroje sú označované ako Dom U. Písmeno U označuje režim „unprivileged" a to z toho dôvodu, že tieto domény nie sú oprávnené kontrolovať hypervizora alebo manipulovať s ostatnými doménami.



Obrázok 11: Architektúra Citrix Hypervisor

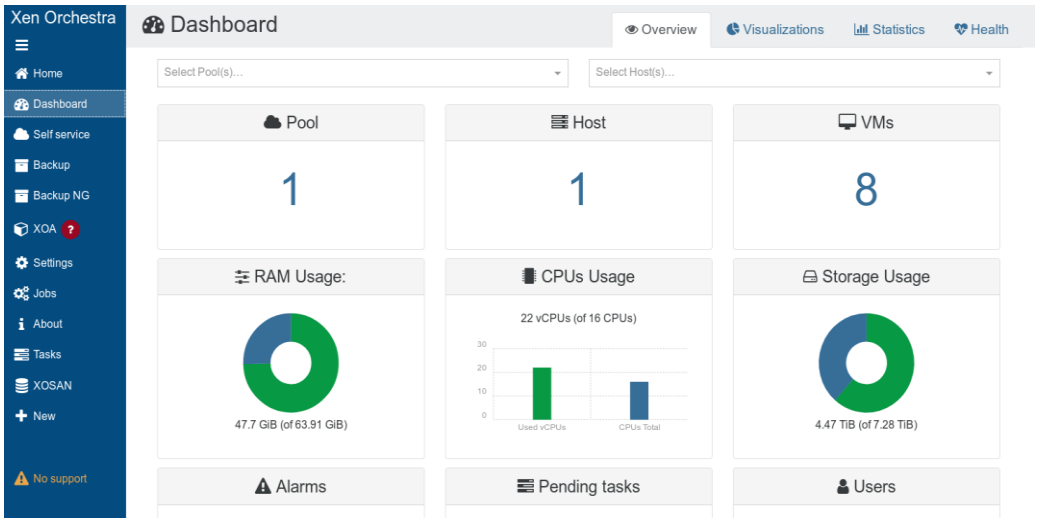
2.2.2 Nástroje pre správu Citrix Hypervisor virtuálnych strojov

Hlavným nástrojom pre správu Citrix Hypervisor je aplikácia s grafickým používateľským rozhraním **XenCenter** v aktuálnej verzií 8.1 a ktorá je dostupná zdarma. XenCenter umožňuje nasadenie virtuálnych strojov, ich konfiguráciu, administráciu, správu virtuálnych sietí alebo zobrazuje výkonnostné štatistiky virtuálnych strojov[18]. Jedná sa o lokálne nainštalovanú aplikáciu dostupnú len pre operačný systém Windows. Minimálne požiadavky pre inštaláciu XenCenter zahŕňajú nutnosť inštalácie na operačný systém Windows a .NET framework vo verzii 4.6 a vyššie. Pre používateľov operačného systému Linux existuje alternatíva v podobe OpenXenManager, ktorá je klonom XenCenter. OpenXenManager bol ale naposledy aktualizovaný ešte v roku 2017, takže mu môžu chýbať niektoré dôležité funkcie z XenCenter.



Obrázok 12: Používateľské rozhranie XenCenter

Druhou alternatívou ako spravovať Citrix Hypervisor je webová aplikácia tretej strany **Xen Orchestra**. Xen Orchestra, v kontraste s XenCenter je založená na operačnom systéme Linux a používa svoj vlastný webový server, ktorý umožňuje správu Citrix Hypervisor infraštruktúry z teoreticky akéhokoľvek operačného systému. Xen Orchestra je funkciami veľmi podobná aplikácií XenCenter, stále pridáva nové funkcie a poskytuje podporu pre spoločnosti, ktoré ju vyžadujú. XOA sa inštaluje priamo na Citrix Hypervisor hostiteľa. Po inštalácii je k dispozícií Xen Orchestra webové rozhranie, cez ktoré je možné spravovať Citrix Hypervisor infraštruktúru[19]. XenOrchestra je dostupná v 3 platených verziách a to Starter, Enterprise a Premium, taktiež ponúka verziu Free, ktorá je zdarma a ponúka základnú funkcionalitu.



Obrázok 13: Používateľské rozhranie Xen Orchestra[19]

2.3 Microsoft Hyper-V

Hyper-V je virtualizačná platforma a hypervizor od spoločnosti Microsoft. Hyper-V bol po prvý krát uvedený na trh v roku 2008 ako súčasť serverového operačného systému Windows Server 2008[20]. Jeho popularita sa rýchlo rozšírila a v dnešnej dobe patrí medzi hlavných konkurentov VMware ESXi. Windows Server v aktuálnej verzií 2019, ktorého súčasťou je aj Hyper-V ponúka dve platené verzie. Jedná sa o verzie Datacenter a Standard. Existuje aj tretia, platená verzia Essentials určená pre malé firmy a podniky s 50 zariadeniami alebo 25 používateľmi, ktorá ale verziou Windows Server 2019 Essentials končí a spoločnosť Microsoft odporúča zákazníkom prechod na riešenie Microsoft 365 Business obsahujúci operačný systém Windows 10 alebo kancelársky balík Office 365[22].

Verzia Standard je ideálnou voľbou pre podniky s malou virtuálnou infraštruktúrou, ktoré požadujú robustné a efektívne riešenie. Jedna licencia verzie Standard umožňuje vytvorenie dvoch virtuálnych strojov a jedného Hyper-V hostiteľa.

Verzia Datacenter je ideálnou voľbou pre podniky s vysokým pracovným zaťažením a veľkou virtuálnou infraštruktúrou. Verzia Datacenter podporuje všetky možnosti verzie Standard a pridáva ďalšie, pre niekoho kľúčové funkcie. V prípade zakúpenia jednej licencie môžeme vytvárať neobmedzené množstvo virtuálnych strojov, ale podobne, ako pri verzií Standard len jedného Hyper-V hostiteľa. Naviac tu oproti verzii Standard nájdeme funkcie ako Shielded Virtual Machine zabezpečujúca dodatočnú bezpečnosť a šifrovanie alebo Software-Defined Networking (SDN), ktorá administrátorovi umožňuje centrálnu správu fyzických a virtuálnych sieťových zariadení[21].

Obe verzie, teda Datacenter a Standard môžu byť nainštalované v dvoch módoch:

- Server Core, neobshuje grafické používateľské rozhranie, ale pre správu hostiteľa využíva len príkazový riadok alebo Windows Powershell.
- Desktop Experience, obsahuje grafické používateľské rozhranie pre správu hostiteľa, veľmi podobné tomu, ktoré poznáme z operačného systému Windows 10.

Hyper-V hypervizor je ponúkaný v troch edíciách:

- Windows Server Hyper-V.
- Hyper-V Server.
- Client Hyper-V.

Windows Server Hyper-V je najrozšírenejšou verziou. Na Windows Server sa nainštaluje Hyper-V ako jeho rola. To znamená, že Hyper-V v podstate nahradí kernel Windows Serveru a správa sa ako operačný systém určený na správu Hyper-V infraštruktúry[20].

Hyper-V Server je voľne dostupná verzia hypervizora Hyper-V. Na rozdiel od Windows Server, Hyper-V Server neponúka grafické používateľské rozhranie, ale len príkazový riadok alebo Windows Powershell a obsahuje základnú funkcionalitu. Táto verzia je ideálna pre malé projekty a testovacie účely.

Client Hyper-V je verzia hypervizora Hyper-V určeného pre desktopové operačné systémy Windows 8 a novšie. Je dostupný pre verzie Windows Pro, Enterprise a Education, nie je dostupný pre verziu Home Edition. Ponúka podobnú funkcionalitu ako Windows Server Hyper-V, ale nie je určený pre produkčné zaťaženie. Client Hyper-V je hlavne používaný ako domáce testovacie prostredie.

2.3.1 Architektúra Microsoft Hyper-V

Hyper-V je hypervizor Typu 1, ktorý využíva hardvérovú virtualizáciu. Medzi používateľmi je často mylne považovaný ako hypervizor Typu 2[23]. Používateľa dokáže ľahko zmiasť fakt, že Hyper-V beží predsa na hostiteľskom serverovom OS Windows Server alebo hostiteľskom desktopovom OS Windows, to je jasná črta práve hypervizora Typu 2. Tu si ale musíme uvedomiť, že Hyper-V beží v týchto operačných systémoch ako jeho rola. Tak ako som už spomenul, keď túto rolu povolíme, Hyper-V v podstate nahradí kernel operačného systému Windows Server alebo desktopového Windows, ktoré potom slúžia na riadenie Hyper-V infraštruktúry.

Root Partition (Koreňová partícia) hlavná partícia, má prístup k hardvérovým prostriedkom serveru. Táto partícia ma na starosti vytváranie podradených partícií, ktoré slúžia k hosťovaniu jednotlivých operačných systémov. Túto hlavnú partíciu predstavuje práve operačný systém Windows Server alebo desktopový Windows.

Child Partition (Podradená partícia) je partícia, ktorá hosťuje jednotlivé operačné systémy. Prístup k fyzickým zdrojom, ako pamäť a zariadenia je child partícií poskytovaná cez **Virtual Machine Bus (VMBus)** alebo hypervizora.

Hypercalls je rozhranie, ktoré slúži pre komunikáciu s hypervizorom.

IC (Integration Component) umožňuje podradeným partíciam komunikáciu s ostatnými partíciami a hypervizorom.

VID (Virtualization Infrastrucutre Driver) poskytuje jednotlivým partíciam služby pre správu virtuálnych procesorov a pamäte.

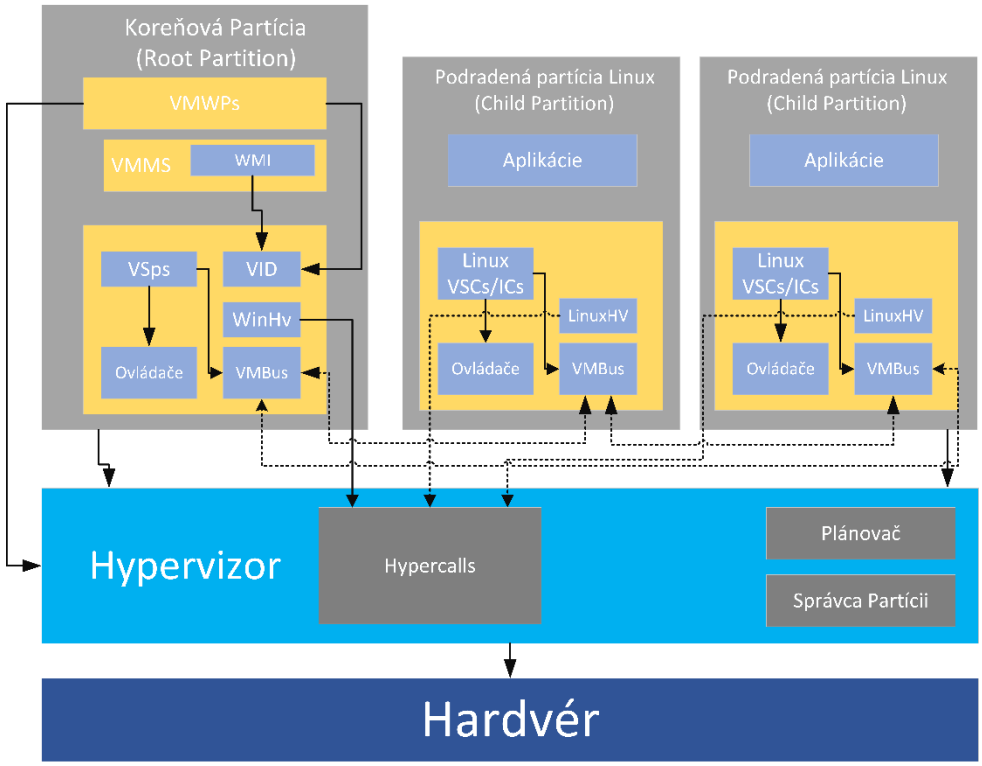
VMMS (Virtual Machine Management Service) je zodpovedná za správu a stav všetkých virtuálnych strojov nachádzajúcich sa v podradených partíciach.

VSC (Virtualization Service Client) sa nachádzia v podradenej partícii, jej úlohou je zaobstarať podradeným partíciam I/O zariadenia, ktoré vyžadujú.

VSP (Virtualization Service Provider) sa nachádza v hlavnej root partícií a prostredníctvom VMBus poskytuje podporu zariadení pre podradené partície.

WinHv/LinuxHv (Hypervisor Interface Library) je spojenie alebo most medzi hypervizorom a ovládačmi jednotlivých operačných systémov, umožňuje zavolať hypervizor v prípade potreby.

WMI (The Virtual Machine Management Service) je API založená na Windows Management Instrumentation (WMI), jej úlohou je správa a kontrola jednotlivých virtuálnych strojov[24].

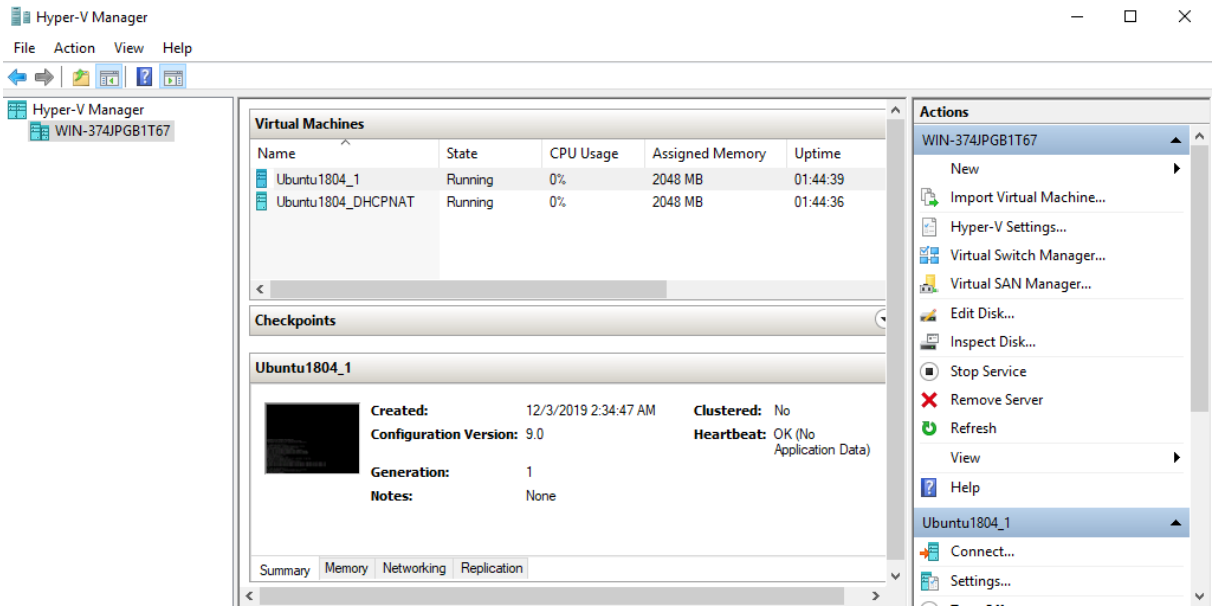


Obrázok 14: Architektúra Hyper-V

2.3.2 Nástroje pre správu Microsoft Hyper-V virtuálnych strojov

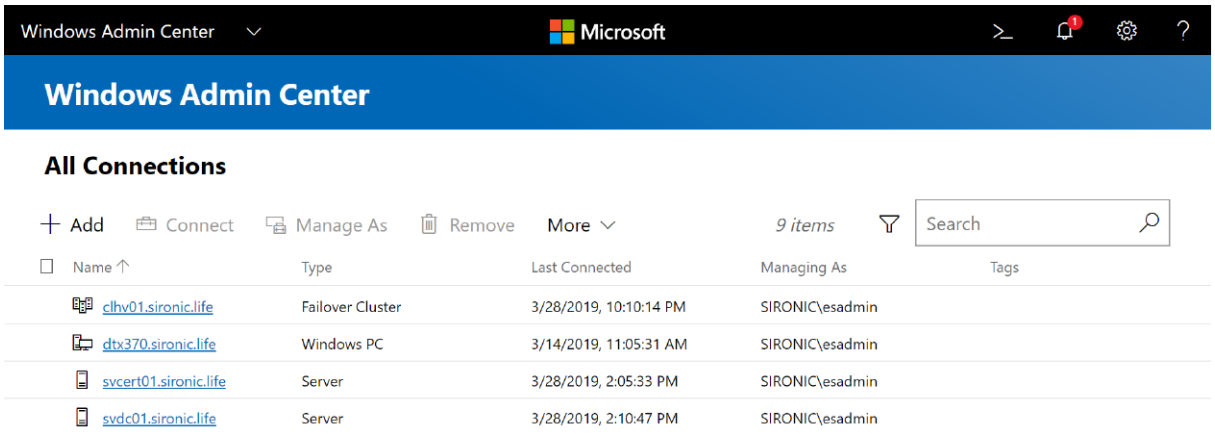
Hyper-V Manager je voľne dostupná aplikácia od spoločnosti Microsoft určená ku správe Hyper-V hostiteľov a jeho virtuálnych hostí, umožňuje lokálnu alebo vzdialenú správu. Poskytuje grafické používateľské rozhranie. Pomocou Hyper-V Managera môžeme vytvárať, konfigurovať alebo vymazať jednotlivé virtuálne stroje, umožňuje konfiguráciu virtuálnych prepínačov, diskov, sieťových adapérov alebo Hyper-V hostiteľa. Hyper-V Manager bol po prvý krát uvedený s operačným sysémom Windows Server 2008. V dnešnej dobe je už súčasťou skoro každého operačného systému Windows, či už sa jedná o Windows Server 2008 a jeho novšie vydania alebo aj pôvodne nepodporovaného staršieho desktopového operačného systému Windows 7, na ktorý je ale nutné doinštalovať Remote Server Administration Tools (RST), ktorého je Hyper-V Manager súčasťou[25]. Keď povolíme Hyper-V rolu na OS Windows Server, tak je automatický nainštalovaný aj Hyper-V Manager. V prípade, že chceme Hyper-V infraštruktúru spravovať vzdialene, napríklad z osobného počítača, tak môžeme Hyper-V Managera povoliť a nainštalovať aj manuálne.

Nevýhodou Hyper-V Managera je to, že je súčasťou operačného systému Windows a teda podobne ako v predchádzajúcej kapitole spomenutý XenCenter nie je dostupný pre iné platformy ako operačný systém Linux alebo macOS. Toto môžeme vyriešiť inštaláciou Windows virtuálneho stroja na osobný počítač s operačným systémom Linux alebo macOS, na ktorom následne povolíme a nainštalujeme Hyper-V Managera. Druhou možnosťou je pripojenie sa prostredníctvom Remote Desktop Protokolu (RDP) k vzdialenej ploche, ale to len v prípade Windows Server Hyper-V nainštalovanom v móde Desktop Experience.



Obrázok 15: Používateľské rozhranie Hyper-V Manager

Windows Admin Center reprezentuje moderný spôsob správy a monitorovania operačných systémov Windows Server a Windows. Ponúka kontrolu nad jednotlivými komponentami, ako hardvérové ovládače. Taktiež ním môžeme spravovať jednotlivé Windows role, príkladom je Hyper-V. Windows Admin Center je v popredí prezentovaný ako moderné a uhladené webové používateľské rozhranie založené na HTML 5. Na pozadí využíva PowerShell ku kontrole jednotlivých systémov. Jeho výhodou je, že na strane klienta nemá žiadne veľké požiadavky, postačuje hocijaký moderný webový prehliadač podporujúci HTML 5. Na druhú stranu, jeho značnou nevýhodou je jeho kompatibilita so staršími operačnymi systémami Windows. Ak ho chceme použiť ako klienta, tak ho musíme nainštalovať na desktopový operačný systém Windows 10 alebo na hocijakú edíciu serverového operačného systému Windows Server 2016 a novšiu. Windows Admin Center dokáže spravovať širšiu škálu operačných systémov Windows, ako tie, na ktoré ho je možné nainštalovať. Medzi podporované systémy patria Windows Server 2008 R2 a novšie a Windows 10. Windows Admin Center má ale nevýhodu v tom, že pokiaľ ho používame ku správe serverových operačných systémov starších ako Windows Server 2019, tak jeho funkcionalita je značne obmedzená[26].



Obrázok 16: Používateľské rozhranie Windows Admin Center[26]

2.4 KVM

Kernel-based Virtual Machine (KVM) je narozdiel od proprietárnych riešení VMware ESXi alebo Microsoft Hyper-V voľne dostupná open source virtualizačná technológia vydávaná ako slobodný softvér pod licenciou General Public License (GPL). KVM je virtualizačná technológia vstavaná priamo do operačného systému Linux. KVM umožňuje prekonvertovanie Linux kernelu do hypervizora, ktorý umožňuje hosťovať viacero izolovaných virtuálnych prostredí. KVM bol uvedený v roku 2006 a od vezie operačného systému Linux 2.6.20 a jeho novších vydaní je jeho súčasťou. Pretože KVM je v podstate časť existujúceho Linux kódu, tak okamžite benefituje z každej novej funkcie, opravy a vývoja Linuxu bez dodatočného inžinierstva. Medzi hlavných prispievateľov do vývoja KVM patrí spoločnosť Red Hat, ktorá KVM využíva vo svojom komerčnom riešení Red Hat Enterprise Linux[27].

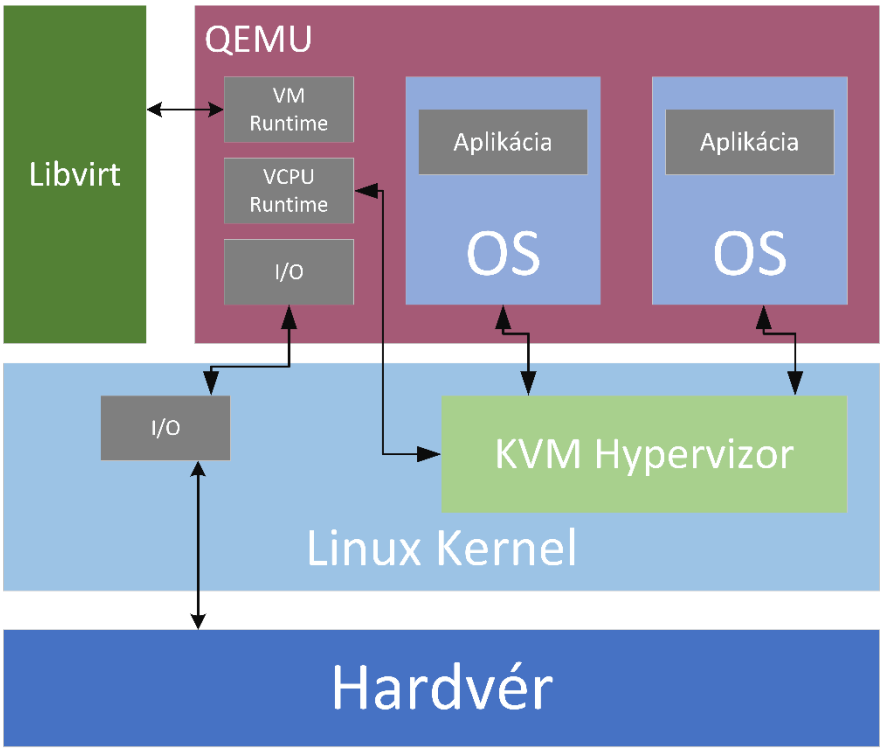
Všetko, čo potrebujeme k implementácií KVM je nutnosť inštalácie na hardvér, ktorý podporuje architektúru x86 a Linux kernel vydaný po roku 2007 bežiaci na procesore od spoločnosti Intel s podporou Intel VT (Virtualization Technology) rozšírení alebo na procesore od spoločnosti AMD s podporou SVM rozšírení, technológia tiež nazývaná ako AMD-V[28]. Tieto rozšírenia sú inštrukčné sady, poskytujúce hardvérovú asistenciu hypervizorovi. KVM podporuje hostiteľa s 64-bit procesorom a 32-bit alebo 64-bit hostí, nepodporuje 64-bitových hostí na hostiteľovi s 32-bit procesorom.

2.4.1 Architektúra KVM/QEMU

Ako teda KVM funguje? KVM prekonvertuje Linux kernel do hypervizora Typu 1 využívajúceho plnú virtualizáciu. Všetky typy hypervizorov potrebujú systémové prostriedky ako správcu pamäte, plánovač procesov alebo ovládače zariadení k tomu, aby mohli spravovať virtuálne stoje. KVM má všetky tieto potrebné komponenty vďaka tomu, že je časť Linux kernelu[27]. Každý jeden virtuálny stroj je implementovaný ako regulárny proces Linuxu, plánovaný štandardným Linux plánovačom, s dedikovaným virtuálnym hardvérom, ako sieťová karta, grafický adaptér, CPU, pamäť a disky.

QEMU (Quick Emulator) je dôležitý komponent, môžeme ho definovať ako hypervizor Typu 2, ktorý vykonáva softvérovú emuláciu diskov, USB portov alebo siete[29]. QEMU dokáže bežať nezávisle na KVM a emulovať všetky prostriedky virtuálnych strojov, ale to, že sa o emuláciu stará softvér ho robí extrémne pomalým. QEMU sa využíva hlavne v spojení s KVM. Kde KVM slúži QEMU ako akcelerátor, tým mu umožňuje využitie virtualizačných rozšírení Intel VT a AMD-V fyzického CPU. Keď to zhrnieme, tak QEMU je Hypervizor Typu 2, ktorý beží v používateľskom prostredí Linuxu a vykonáva emuláciu hardvéru, na druhej strane KVM je Hypervizor Typu 1, ktorý beží v prostredí Linux Kernelu a umožňuje používateľskému prostrediu prístup k hardvérovým prostriedkom.

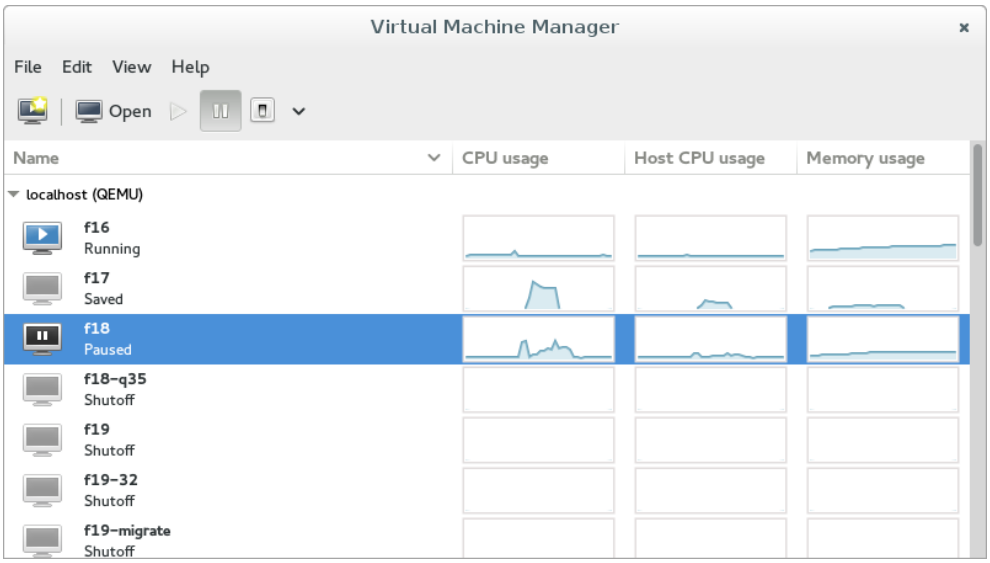
Libvirt je súbor softvéru, ktorý poskytuje pohodlnú správu virtuálnych strojov a inej virtualizačnej funkcionality, ako správu úložiska a sieťových rozhraní. Zahŕňa dlhodobo stabilné C API, démona libvirtd a program pracujúci v príkazovom riadku virsh. Hlavným cieľom libvirt je poskytnutie jednotnej cesty ako spravovať viacero hypervizorov, ako KVM/QEMU alebo LXC[30].



Obrázok 17: Architektúra KVM

2.4.2 Nástroje pre správu KVM/QEMU virtuálnych strojov

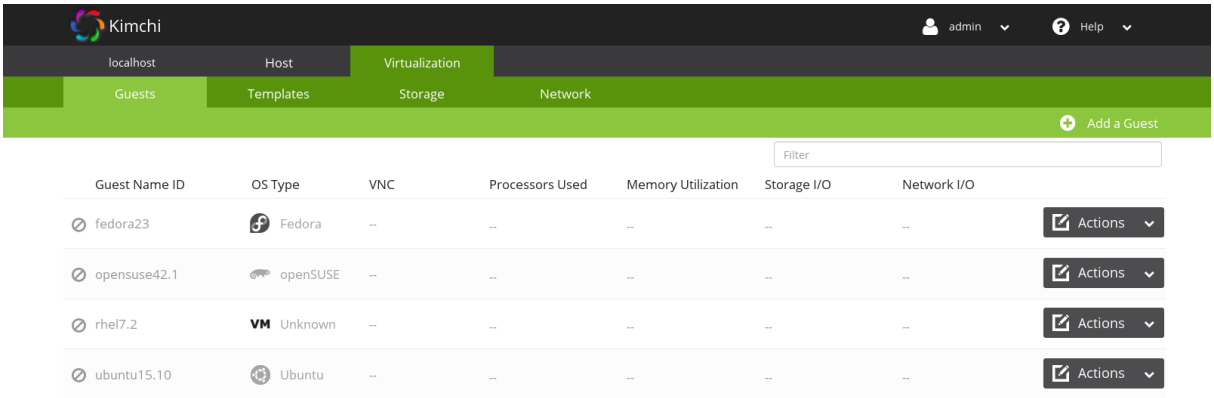
Virt-manager (Virtual Machine Manager) je desktopová aplikácia pre operačný systém Linux, ktorá ako iné slúži na lokálnu alebo vzdialenú správu, vytváranie, spúšťanie, zmazanie virtuálnych strojov, správu virtuálnych sietí a podobne. Virt-manager obsahuje jednoduché grafické používateľské rozhranie pre správu virtuálnych strojov cez libvirt. Je primárne určený pre virtuálne stroje hosťované prostredníctvom KVM, ale dokáže spravovať aj Xen alebo linuxové kontajnery (LXC). Dokáže zobraziť aktuálny výkon a štatistiky využitia zdrojov jednotlivých virtuálnych strojov[31].



Obrázok 18: Používateľské rozhranie virt-manager[31]

Virsh je nástroj pracujúci v príkazovom riadku a neponúka grafické používateľské rozhranie. Umožňuje administrátorovi vytvorenie, zmazanie, spúšťanie, vypínanie a konfiguráciu jednotlivých virtuálnych strojov. Virsh je obzvlášť užitočný pre skúsených Linux administrátorov, ktorí sa zaujímajú o skriptovanie alebo automatizáciu niektorých aspektov riadenia svojich virtuálnych strojov. Virtuálne stroje spravované pomocou virsh sú vytvorené opisom virtuálneho stroja v libvirt XML súbore a následným importovaním tohto súboru do virsh[32]. Pomocou virsh môžeme virtuálne stroje spravovať lokálne alebo vzdialene, pripojením sa na hostiteľa napríklad prostredníctvom SSH (Secure Shell). Takže toto riešenie je možné využiť aj pre operačné systémy Windows alebo macOS.

Kimchi je aplikácia pre správu KVM virtuálnych strojov prostredníctvom libvirt. Používa moderné webové používateľské rozhranie založené na HTML 5 dostupné z každého moderného aktualizovaného webového prehliadača. Virsh a virt-manager sú aplikácie, ktoré môžu byť pre neskúseného Linux administrátora zo začiatku neprehľadné a zložité, hlavne v prípade virsh. Tomuto problému sa snaží predísť Kimchi, ktoré je navrhnuté tak, aby administrátorovi čo najviac uľahčilo prácu s KVM pri vytváraní viruálnych strojov. Kimchi beží na Wok plugine, vyvíjanom Kimchi vývojármi. Je založený na cherrypy webovom frameworku s podporou HTML 5. Wok je nevyhnutnou súčasťou Kimchi a musí byť nainštalovaný na KVM hostiteľa pred použitím Kimchi. Kimchi podporuje väčšinu distribúcií operačného systému Linux, ale dôraz je kladený na testovanie na distribúciach Ubuntu, Fedora a openSUSE[33].



Obrázok 19: Používateľské rozhranie Kimchi[33]

2.5 Linux kontajnery

Linux kontajner je skupina jedného alebo viacerých procesov, ktoré sú izolované od zvyšku základného operačného systému Linux. Všetky súbory potrebné pre beh kontajnera sú poskytované zo separátneho obrazu. Takýto obraz kontajneru a jeho obsah môžeme v podstate považovať ako inštaláciu Linuxovej distribúcie, pretože obsahuje kompletné inštalačné balíčky a konfiguračné súbory. Niekto by si mohol myslieť, že sa predsa jedná o klasickú virtualizáciu, to ale nie je úplne pravda. Kontajnery, na rozdiel od klasickej virtualizácie zdieľajú rovnaké jadro hostiteľského operačného systému Linux a izolujú aplikačné procesy od zvyšku systému[34].

Tabuľka 1: Porovnanie Linux kontajneru a Virtuálneho stroja

Linux kontajner	**Virtuálny stroj**
Využíva virtualizáciu na úrovni OS	Využíva plnú virtualizáciu
Kontajnery zdieľajú hostiteľský OS a jeho jadro	Každý VM používa svoj vlastný OS
Menej náročné na pamäť a CPU	Celkovo náročnejšie na pamäť a CPU
Rýchly štart systému	Pomalý štart systému
Čiastočná izolácia na úrovni procesu, menej bezpečné a stabilné	Plne izolované, bezpečnejšie a stabilnejšie
Veľkosť obrazu v desiatkách MB	Veľkosť obrazu OS v jednotkách GB
Technológie: LXC/LXD, Docker	Technológie: VMware ESXi, Microsoft Hyper-V, KVM

2.5.1 LXC

LXC je používateľské rozhranie, ktoré využíva možnosti jadra operačného systému Linux. Prostredníctvom výkonnej API a jednoduchých nástrojov umožňuje úžívateľom ľahké vytváranie a správu systémových alebo aplikáčných kontajnerov[36]. Medzi hlavné vlastnosti jadra Linuxu, ktoré LXC využíva patria:

- **Namespaces jadra**, sú funkciou, ktorá kontajneru umožňuje správanie a prezentovanie sa ako úplne samostatný virtuálny stroj[37]. Existuje 7 rôznych „namespaces", z ktorých každá izoluje svoju vlastnú inštanciu:
  Cgroups - izoluje koreňový adresár.
  IPC - izoluje komunikáciu medzi procesmi.
  Network - izoluje sieť, zariadenia, porty.
  Mount - izoluje úložný priestor.
  PID - izoluje ID procesov.
  User - izoluje ID používateľa a skupiny.
  UTS - izoluje doménové a hostiteľské mená.
- **Profily Apparmor a SELinux**.
- **Seccomp politika**.
- **Chroots**.
- **Schopnosti jadra**, práva koreňového používateľa, ktoré môžu byť povolené alebo zablokované pre konkrétny proces.

LXC kontajnery sú často považované za niečo medzi chroot (change root) a klasickým virtuálnym strojom. LXC bolo vytvorené s cieľom možnosti vytvorenia virtúalneho prostredia, ktoré sa čo najviac blíži štandardnému virtuálnemu stroju, ale bez nutnosti potreby samostatného jadra, pre každý z nich. Typickým zástupcom, ktorý rozširuje možnosti LXC je Docker alebo LXD.

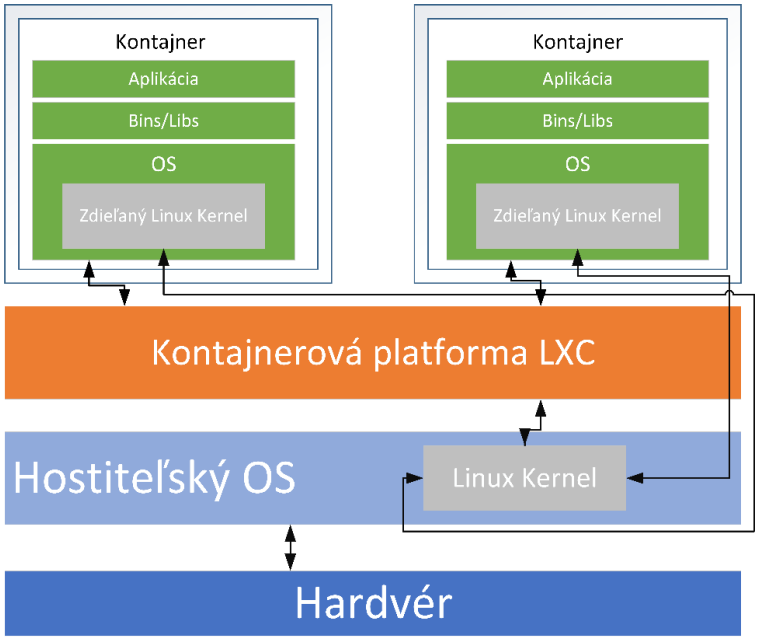
Hlavné komponenty, ktoré tvoria LXC sú: liblxc knižnica, niekoľko prepojení programovacích jazykov a API rozhrania (python3, lua, Go, ruby, Haskell), sada nástrojov pre správu kontajnerov, šablóny pre distribúciu kontajnerov[36].

LXC kontajnery delíme na dva typy, ktoré hrajú dôležitú úlohu v prípade bezpečnosti:

- Privilegované kontajnery.
- Neprivilegované kontajnery.

Privilegované kontajnery sú definované ako kontajnery, ktorých uid 0 je mapované na uid 0 hostiteľa. V takýchto kontajneroch je bezpečnosť hostiteľa čisto v režií Apparmor, SELinux alebo namespaces. Toto zabezpečenie vie vo väčšine prípadov predísť poškodeniu hostiteľa, takéto poškodenie môžeme definovať ako rekonfiguráciu hostiteľského hardvéru, hostiteľského kernelu alebo nežiadúceho pístupu do súborového systému hostiteľa[38]. Tieto kontajnery nie sú vo všeobecnosti považované za bezpečné.

Neprivilegované kontajery sú „safe by design"[38] a boli vytvorené za účelom žvýšenia bezpečnosti. Uid 0 je mapované na neprivilegovaného užívateľa mimo kontajera. S takýmto kontajnerom odpadá nutnosť zabezpečenia pomocou Apparmor alebo SELinux.



Obrázok 20: Architektúra LXC

2.5.2 LXD

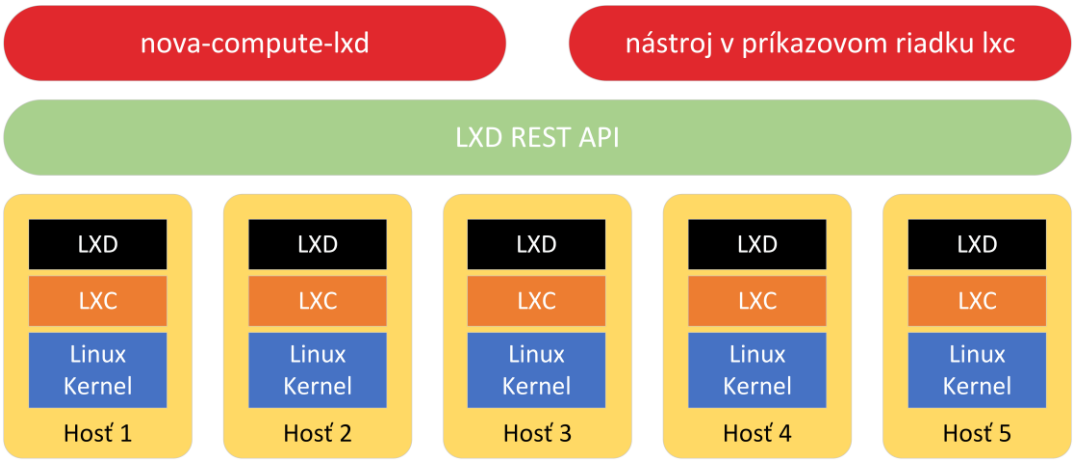
LXD (lex-dee) môžeme jednoducho definovať ako rozšírenie LXC, nie jeho upravenú verziu. LXD vlastne stavá na základoch LXC a poskytuje nový, lepší používateľský zážitok z celkového používania. Spoločnosť Canonical, ktorá vyvíja jednú z najznámejších Linux distribúcii Ubuntu a ktorá je jedna z hlavných podporovateľov LXC uviedla v roku 2014 aj LXD[39]. LXD je správca novej generácie určený na správu systémových kontajnerov. Ponúka podobnú používateľskú funkcionalitu, akú poznáme zo štandardných virtuálnych strojov, ale pracuje s Linux kontajnermi. Je založený na predpripravených obrazoch širokej škály rôznych distribúcií operačného systému Linux a je postavený okolo jednoduchej, ale výkonnej REST (Representational State Transfer) API a pomocou knižnice liblxc komunikuje s LXC[40].

Medzi najväčšie výhody LXD patrí:

- „Safe by design", neprivilegované kontajnery.
- Intuitívne používateľské rozhranie lxc pracujúce v príkazovom riadku.
- Založený na predpripravených obrazoch širokej škály distribúcii OS Linux, ktoré sú denne aktualizované.
- Ponúka pokročilú kontrolu systémových prostriedkov ako CPU, pamäť alebo úložisko.
- Možnosť správy siete, napríklad vytváranie a konfigurácia mostov.
- Dobrá škálovateľnosť.

LXD je voľne dostupný softvér vytvorený pomocou programovacieho jazyka Go a vyvíjaný pod licenciou Apache 2. LXD je vydávaný v dvoch verziách:

- LTS (Long Term Support) - je dlhodobo podporovaná verzia, ktorá dostáva pravidelné bezpečnostné aktualizácie a rôzne opravy chýb. Aktuálna verzia je LXD 3.0, ktorá je podporovaná do Júna 2023.
- Mesačné vydania - sú vydávané každý mesiac a obsahujú nové funkcie a opravy chýb. Podpora pre tieto verzie je veľmi krátka, väčšinou sa jedná o maximálne mesiac. Aktuálna verzia k 29.1.2020 je LXD 3.19.



Obrázok 21: Architektúra LXD[41]

2.5.3 Porovnanie LXC/LXD s platformou Docker

Kontajnery môžeme klasifikovať do dvoch skupín, podľa virtualizácie, ktorú využívajú:

- Plná systémová virtualizácia - LXC/LXD.
- Virtualizácia na úrovni aplikácie - Docker.

Pri plnej systémovej virtualízacií, ktorú využíva LXC/LXD si môžeme vybrať obraz preferovaného operačného systému Linux, na ktorý nainšalujeme požadované služby a aplikácie, týkajúce sa napríklad webu alebo e-mailu. Virtualizácia na úrovni aplikácie, ktorú využíva Docker je zameraná na jednu konkrétnu aplikáciu alebo službu. LXC/LXD kontajery môžu byť teda použité ako virtuálne prostredia, s podobnou funkcionalitou, akú poznáme pri klasických virtuálnych strojoch. Pre vývoj aplikácií, kde sa požaduje rýchle testovanie a nasadenie konkrétnej aplikácie je zasa vhodnejší Docker[42]. Keď si uvedomíme tieto rozdiely, tak prídeme na to, že po pár úpravách týkajúcich sa hlavne bezpečnosti, nám vlastne nič nebráni v tom, aby sme mali vo vnútri LXC/LXD kontajneru Docker kontajner[43].

Tabuľka 2: Porovnanie LXC/LXD s Docker

LXC/LXD	**Docker**
Plná systémová virtualizácia	Virtualizácia na úrovni aplikácie
LXD démon dostupný len pre OS Linux, klient aj pre OS Linux, Windows a macOS	Kompletne dostupný pre OS Linux, Windows a macOS
Môžeme vykonávať zmeny vo vnútri spusteného kontajnera - Stavový	Zmeny kontajnera na úrovni obrazu, po spustení nemôžeme vykonávať zmeny - Bezstavový
Viacúčelový	Jednoúčelový
Menej bezpečný - Malvér sa môže rozšíriť medzi ostatné aplikácie bežiace v rovnakom prostredí	Bezpečnejší - Aplikácia beží vo svojom vlastnom izolovanom prostredí

2.5.4 Nástroje pre správu LXC/LXD virtuálnych prostredí

Jadrom LXD, vývojármi niekedy nazývaného aj kontajnerový „lightervisor" je démon, ktorý prostredníctvom REST API kontroluje systémové kontajnery. Pre správu využíva dva typy klientov:

- **lxc klient** (Nepliesť s LXC) je nástroj pracujúci v príkazovom riadku, zameraný na správu malej až strednej infaštruktúry, od jedného hostiteľa, až po niekoľko desiatok hostiteľov a ich hostí.
- Modul OpenStack Nova, nazývaný **nova-compute-lxd** je zameraný na bezproblémovú integráciu s OpenStack infraštruktúrou.

Najjednoduchším a určite i najrozšírenejším spôsobom, ako spravovať LXD infraštruktúru je pomocou lxc nástroja pracujúceho v príkazovom riadku. Predtým, ako začneme spravovať LXD démona prostredníctvom lxc klienta musíme pridať užívateľov, ktorí sú oprávnení spravovať LXD do takzvanej lxd skupiny (root uzívateľ je pridaný automaticky). Tákýto užívatelia majú plnú kontrolu nad LXD infraštruktúrou.

Klient lxc štandardne pracuje s lokálne nainštalovaným démonom, ale umožňuje aj vzialenú správu viacerých LXD démonov, ktoré môžeme jednoducho pridať. Samostatný klient lxc je dostupný pre operačné systémy OS Linux, Windows a macOS. Ďalšou možnosťou je podobne ako pri virsh vzdialené pripojenie pomocou napríklad SSH na vzdialený server. Konkrétnymi príkazmi a využitím lxc klienta som sa zaoberal v praktickej časti. Existujú aj ďalšie možnosti správy LXD, najznámejšou alternatívou je pravdepodobne **lxdui**, ponúka webové grafické používateľské rozhranie, ktoré ale podľa používateľov obsahuje veľa chýb a posledné dva roky dokonca ani nebolo aktualizované[44].

3 Možnosti využitia grafických kariet pre virtualizáciu

Na tradičnom fyzickom zariadení ako stolný počítač alebo prenosný počítač, vykonáva GPU (Graphical Processing Unit) náročné komplexné úlohy spojené s 3D aplikáciami alebo videom. V počiatkoch virtualizácie boli tieto náročné úlohy vykonávané hostiteľským CPU, ktoré je schopné dosiahnuť požadovanú funkcionalitu pre niektoré základné aplikácie, ale nikdy nedosahoval výkonnosť vyžadovanú používateľmi. Zmena nastala pred niekoľkými rokmi s príchodom virtuálnych GPU (vGPU). Virtualizácia GPU napríklad v data centere umožnila rozdelenie vGPU medzi jednotlivé virtuálne stroje, čo rapídne vylepšilo výkonnosť pre aplikácie a umožnilo organizáciam vybudovať virtuálne desktopové infraštruktúry (VDI) .

Čo je vlastne účelom GPU? 3D aplikácie, video a vykreslenie obrázkov sú všetko masívne paralelné úlohy, pre ktoré je GPU obsahujúce tisícky výpočtových jadier ideálne riešenie. Inžinieri sa na ne spoliehajú pri náročných úlohách, ako počítačovo podporované inžinierstvo (CAE), počítačovo podporovaný dizajn (CAD) alebo počítačovo podporovaný priemysel (CAM). Samozrejme aj procesor, ktorý je ale vhodný pre sériové úlohy môže vykreslovať grafiku, 4, 8, 16 jadrové procesory by túto úlohu mohli zvádnuť, ale na to tu už máme ďaleko efektívnejšie GPU[45].

3.1 NVIDIA vGPU

V prípade spoločnosti NVIDIA je ich riešenie vGPU tvorené softvérom. NVIDIA vGPU softvér poskytuje grafické virtuálne prostredia akcelerované najvýkonnejšími datacentrovými GPU NVIDIA Tesla. Vo VDI prostredí poháňanom NVIDIA vGPU je softvér nainštalovaný na virtualizačnú vrstvu hostiteľa spoločne s hypervizorom. Tento softvér vytvorí virtuálne GPU, ktoré môžu byť pridelené medzi jednotlivé virtuálne stroje, jednému virtuálnemu strojmu je možné prideliť aj viacero virtuálnych GPU. Tieto stroje teda zdieľajú fyzickú GPU serveru[46]. NVIDIA vGPU využíva takzvaný „time-slicing", čo to vlastne znamená vysvetlím na jednoduchom príklade. Predstavme si, že máme dva virtuálne stroje, ktoré zdieľajú jednu fyzickú GPU s 1000 jadrami, plánovač pridelí na 50% času celých 1000 GPU jadier jednému virtuálnemu stroju, keby máme jeden virtuálny stroj, tak je to 100%. Môžeme povedať, že je tu aplikovaný model férového zdieľania[48], takže každý virtuálny stroj dostane plných 1000 GPU jadier na zlomok sekundy. Toto plánovanie je vykonávané priamo na fyzickej GPU servera. NVIDIA vGPU taktiež poskytuje grafický ovládač pre každý jeden virtuálny stroj, čo umožňuje každému z nich využívať benefity GPU tak, ako v prípade klasického stolného počítača.

NVIDIA vGPU ponúka štyri rôzne edície[47]:

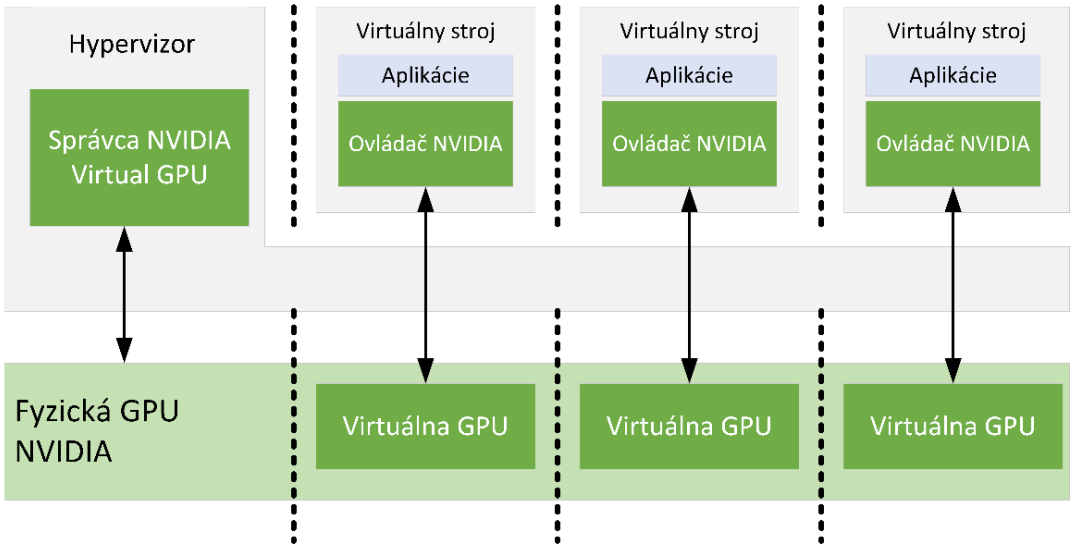
- NVIDIA GRID Virtual PC (GRID vPC) - riešenie, ktoré je ideálne pre používateľov vo VDI, ktorí využívajú hlavne štandardné aplikácie, ako internetový prehliadač alebo multimédiá.
- NVIDIA GRID Virtual Applications (GRID vApps) - pre organizácie využívajúce Citrix XenApp alebo VMware Horizon.
- NVIDIA Quadro Virtual data Center Workstation (Quadro vDWS) - pre vysoké zaťaženie a pre dizajnérov používajúcich náročné 3D aplikácie, ako Siemens NX alebo Autodesk Maya.
- NVIDIA Virtual Compute Server (NVIDIA vComputeServer) - pre organizácie s vysokým výpočtovým zaťažením serverov, ako umelá inteligencia alebo strojové učenie.

NVIDIA vGPU softvér môže byť použitý niekoľkými spôsobmi.

NVIDIA vGPU, umožňuje viacerým virtuálnym strojom využívať jednu fyzickú GPU, podporovaný je Citrix Hypervisor alebo VMware vSphere/ESXi, nie je podporovaný napríklad Microsoft Hyper-V.

GPU Pass-Through, v tomto móde je celá fyzická GPU pridelená jednému virtuálnemu stroju[49]. Podporovaný je Citrix Hypervisor, VMware vSphere/ESXi alebo Microsoft Hyper-V.

Nasadenie ako „bare metal", nasadenie priamo na server bez hypervizora, dostupné pre edície Quadro vDWS a GRID vApps.



Obrázok 22: Architektúra NVIDIA vGPU

3.2 AMD MxGPU

Ďalšou veľkou spoločnosťou na poli GPU je nepochybne AMD. V roku 2016 AMD predstavila svoje riešenie virtualizácie GPU, AMD Multiuser GPU (MxGPU). AMD MxGPU má byť priamým konkurentom pre NVIDIA vGPU, ale obe riešenia se podstatne líšia. Zatiaľ čo NVIDIA vGPU spolieha na softvérovú virtualizáciu, tak AMD MxGPU je hardvérovo založené riešenie. To znamená, že celá virtualizácia je vykonávaná priamo na samotnej GPU a na rozdiel od NVIDIA vGPU nevyžaduje žiadne dodatočné licencie[50].

AMD MxGPU umožňuje viacero virtuálnym strojom priamy prístup k časti fyzickej GPU servera. K tomu využíva takzvanú Single Root I/O virtualizáciu (SR-IOV), na rozdiel od NVIDIA vGPU, ktorá softvérovo rozdelí fyzickú GPU. Vďaka použitiu SR-IOV PCIe virtualizačnému štandardu eliminuje použitie proprietárneho, komplexného a drahého softvéru na strane hypervizora. Každý jeden virtuálny stroj využíva natívne AMD ovládače s plnou kompatabilitou a prístupom k fyzickej GPU serveru[51]. Na rozdiel od NVIDIA vGPU, ktorá využíva „time-slicing", AMD MxGPU pridelí určitý počet GPU jadier každému virtuálnemu stroju. Jednoduchý príklad, máme fyzickú GPU s 1000 jadrami a 10 virtuálnych strojov, to znamená, že každý virtuálny stroj dostane pridelených 100 GPU jadier[50]. Podľa tvrdenia AMD by takéto riešenie malo teoreticky podávať lepšie výkony ako NVIDIA vGPU, pretože je viac úloh spracovávaných hardvérovo.

AMD MxGPU momentálne podporuje VDI prostredia postavené na VMware vSphere/ESXi a Citrix Hypervisor.

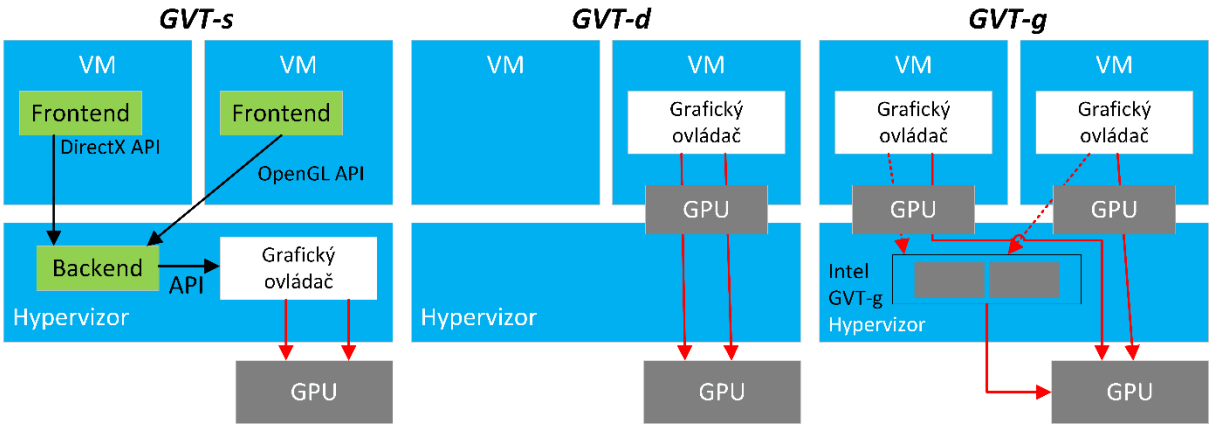
3.3 Intel GVT

Graphics Virtualization Technology (GVT) je riešením od spoločnosti Intel. Od dvoch už spomenutých riešení NVIDIA vGPU, AMD MxGPU sa odlišuje v tom, že nevyžaduje žiaden dodatočný hardvér v podobe samostatnej grafickej karty. Používa Intel Processor Graphics integrovanú priamo v Intel procesoroch a štandardný grafický ovládač Intel GPU v rámci virtuálneho stroja. Portfólio Intel GVT zahŕňa tri odlišné prístupy virtualizácie GPU, jedná sa o GVT-s, GVT-d a GVT-g. Každá z týchto techník bola navrhnutá tak, aby spĺňala špecifické požiadavky používateľa.

Intel GVT-s, virtuálna zdieľaná grafická akcelerácia, viacero virtuálnych strojov zdieľa jednu fyzickú GPU, známe tiež ako zdieľaný virtuálny grafický adaptér (vSGA), odosielanie na základe API (API forwarding). Tento prístup virtualizácie vyžaduje virtuálny grafický ovládač vo vnútri virtuálneho stroja, ten následne umožní odosielanie na základe API. GVT-s podporuje veľké množstvo virtuálnych strojov, ale tento prístup obmedzuje kompatabilitu, výkonnosť a všestrannosť GPU.

Intel GVT-d, virtuálna dedikovaná akcelerácia, jeden virtuálny stroj využíva celú fyzickú GPU, známe tiež ako priamy virtuálny grafický adaptér (vDGA), technológia podobná NVIDIA GPU Pass-through.

Intel GVT-g, využíva virtuálnu GPU (vGPU), viacero virtuálnych strojov zdieľa jednu fyzickú GPU, technológia podobná NVIDIA vGPU. Každý virtuálny stroj využíva kópiu natívneho Intel ovládača a podobne ako NVIDIA vGPU využíva „time-slicing", kde agent priamo v hypervizorovi pridelí plný výkon GPU každému jednému vituálnemu stroju na zlomok sekundy. Intel vyvíja dve GVT-g open-source implemetácie existujúce pre KVM, ktorú nazval KVMGT a pre Xen, XenGT. Podporujú až sedem paralelných virtuálnych GPU[52][53].



Obrázok 23: Architektúra GVT-s, GVT-d a GVT-g

4 Voľba serverových virtualizačných platforiem

Pred samotnou inštaláciou, konfiguráciou a testovaním bolo najprv nutné zvoliť jednotlivé virtualizačné platformy a popísať hardvérové a softvérové prostriedky, ktoré boli v práci použité.

V práci som sa rozhodol porovnať a otestovať niektoré najpoužívanejšie a najrozšírenejšie virtualizačné platformy, ktoré sú momentálne dostupné na trhu. Jedná sa o 4 riešenia využívajúce hypervizor Typu 1 a jednu platformu zastupujúcu Linux kontajnery, konkrétne:

- **VMware ESXi 6.7.0**.
- **Citrix Hypervisor Express Edition 8.0.0**.
- **Microsoft Windows Server Hyper-V 2019**.
- **KVM / Ubuntu 18.04.3 LTS**.
- **LXD 3.0.3 / Ubuntu 18.04.3 LTS**.

4.1.1 Hardvérové prostriedky

K testovaniu boli využité hardvérové prostriedky, ktoré mi poskytla Katedra telekomunikačnej techniky. Mal som k dispozícií gigabitový prepínač Cisco SG350 a 5x Dell PowerEdge R230 1U server s nasledujúcou konfiguráciou:

- **Procesor:** 1x Intel Xeon E3-1220 v5 / Frekvencia: 3GHz / Počet jadier: 4.
- **Operačná pamäť:** 2x 8GB DDR4 2133MHz.
- **Pevný disk:** 2x Seagate ST1000NM0085 / Kapacita: 1000GB.
- **Sieťová karta:** 1x Broadcom BCM5720 NetXtreme Dual-Port 1GBASE-T.

4.1.2 Prehľad pridelených IP adries

Pridelené IP adresy, ktoré som mal k dispozícií na jednotlivých rozhraniach sieťových kariet možeme vidieť v Tabuľke 3. V práci som využíval IP adresy verzie 4.

Tabuľka 3: Prehľad pridelených IP adries a ich rozdelenie

Rozdelenie Serverov	**Rozhranie 1** **MAC/IPv4/IPv6**	**Rozhranie 2** **MAC/IPv4/IPv6**
Server 5 - VMware ESXi 6.7.0	50:9a:4c:88:9d:77 158.196.20.223 2001:718:1001:2c8::223	50:9a:4c:88:9d:78 158.196.20.224 2001:718:1001:2c8::224
Server 4 - Citrix Hypervisor 8.0.0	50:9a:4c:83:7c:6f 158.196.20.226 2001:718:1001:2c8::226	50:9a:4c:83:7c:70 158.196.20.227 2001:718:1001:2c8::227
Server 3 - Microsoft Server 2019 Hyper-V	50:9a:4c:88:9d:8f 158.196.20.228 2001:718:1001:2c8::228	50:9a:4c:88:9d:90 158.196.20.229 2001:718:1001:2c8::229

Server 2 - KVM / Ubuntu 18.04.3 LTS	50:9a:4c:83:79:37 158.196.20.231 2001:718:1001:2c8::231	50:9a:4c:83:79:38 158.196.20.238 2001:718:1001:2c8::238
Server 1 - LXD 3.0.3 / Ubuntu 18.04.3 LTS	50:9a:4c:83:76:3f 158.196.20.233 2001:718:1001:2c8::233	50:9a:4c:83:76:40 158.196.20.234 2001:718:1001:2c8::234

4.1.3 **Operačný systém virtuálnych strojov a prostredí**

Základ všetkých hosťovaných virtuálnych strojov/prostredí tvoril operačný systém Linux a to distribúcia **Ubuntu Server 18.04.3 LTS** „**Bionic Beaver**" s dlhodobou garantovanou podporou do Apríla 2023.

5 Inštalácia a konfigurácia serverových virtualizačných platforiem

V tejto kapitole som sa zaoberal inštaláciou serverových virtualizačných platforiem, vytvorením a konfiguráciou jednotlivých virtuálnych strojov a prostredí, konfiguráciou siete a celkovou prípravou pred samotným testovaním.

5.1 Tvorba Linux brány na základe Ubuntu Server 18.04.3 LTS

K zabezpečeniu internetovej konektivity pre virtuálne stroje vo vnútri virtuálnej siete som si vytvoril virtuálny stroj založený na Ubuntu Server 18.04.3 LTS, ktorý slúžil ako brána do internetu. VMware ESXi, Citrix Hypervisor a Microsoft Server Hyper-V neponúkajú žiaden automatický NAT a DHCP server pre svojich hostí, takže tento postup bol identický pre všetky tri spomenuté riešenia, jediným rozdielom boli názvy sieťových rozhraní, na tento fakt som vždy upozornil pri konkrétnej platforme. Takto vytvorený virtuálny stroj, slúžiaci ako brána do internetu pre ostatné virtuálne stroje som v rámci spomenutých riešení VMware ESXi, Citrix Hypervisor a Microsoft Server Hyper-V pomenoval vždy ako Ubuntu_DHCPNAT, na tomto stroji neboli vykonávané žiadne porovnávacie testy. Pre platformy KVM a LXD bol využitý iný spôsob zabezpečenia internetovej konektivity pre virtuálne stroje, ktorý bol popísaný pri konkrétnej platforme.

Ako prvé som si vytvoril DHCP server inštaláciou balíčka **isc-dhcp-server**:

- `sudo apt update && sudo apt install isc-dhcp-server`

Po nainštalovaní bolo nutné upraviť konfiguračný súbor isc-dhcp-server:

- `sudo nano /etc/default/isc-dhcp-server`

Kde jedinou zmenou bola špecifikácia vnútorného sieťového rozhrania slúžiaceho k obsluhe DHCP žiadostí, v mojom prípade **INTERFACESv4** = "nazovrozhrania".

Ďalšou zmenou bola úprava konfiguračného súboru dhcpd.conf:

- `sudo nano /etc/dhcp/dhcpd.conf`

Tu som si pomocou parametrov špecifikoval svoju vnútornu sieť, kompletný výpis súboru dhcpd.conf sa nachádza v priloženej prílohe A:

```
subnet 192.168.1.0 netmask 255.255.255.0 {
option routers 192.168.1.254;
option subnet-mask 255.255.255.0;
range 192.168.1.100 192.168.1.200;}
```

Následne som spustil rozhranie a pridelil IP adresu a masku brány, ktorú som špecifikoval v súbore dhcpd.conf parametrom option routers, respektíve option subnet-mask:

- `sudo ip link set dev nazovrozhrania up`
  `sudo ip a a IP-adresa/24 dev nazovrozhrania`

Nasledoval reštart DHCP služby a prípadné overenie, či služba nabehla korektne:

- `sudo systemctl restart isc-dhcp-server`

  `systemctl status isc-dhcp-server`

Akonáhle som nakonfiguroval DHCP server, vykonal som konfiguráciu prekladu sieťových adries NAT, predtým som ale povolil presmerovanie portov úpravou súboru sysctl.conf:

- `sudo nano /etc/sysctl.conf`

V tomto súbore bolo nutné odkomentovať alebo dopísať riadok **net.ipv4.ip_forward=1**. Príkazom sudo sysctl -p /etc/sysctl.conf som bez nutnosti reštartovania virtuálneho stroja povolil tieto zmeny.

Po povolení presmerovania som pomocou príkazu pridal NAT pravidlo do iptables:

- `sudo iptables -t nat -A POSTROUTING -o -nazovrozhrania -j MASQUERADE`

Toto pravidlo umožní klientom vo vnútornej sieti pripojenie do internetu. Ďalej som si nainšaloval balíček **iptables-persistent**, ktorý zabezpečil, že pravidlá iptables budú trvalé a nezmažú sa po reštarte virtuálneho stroja:

- `sudo apt install iptables-persistent`

  `sudo iptables-save > /etc/iptables/rules.v4`

Vytvoril som si súbor rc.local, kde som vložil riadok /sbin/iptables-restore < /etc/iptables/rules.v4:

- `sudo nano /etc/rc.local`

S nakonfigurovanou bránou posyktujúcou DHCP a NAT služby som následne vykonal konfiguráciu klientov. Ku konfigurácií som namiesto staršieho riešenia /etc/network/interfaces použil modernejší spôsob Netplan, ktorý využíva k opisu sieťových rozhraní súbor .yaml:

- `sudo nano /etc/netplan/50-cloud-init.yaml`

V 50-cloud-init.yaml súbore som špecifikoval rozhranie, tak aby mu bola automaticky pridelená IP adresa od DHCP serveru, ďalej som špecifikoval IP adresy DNS serverov.

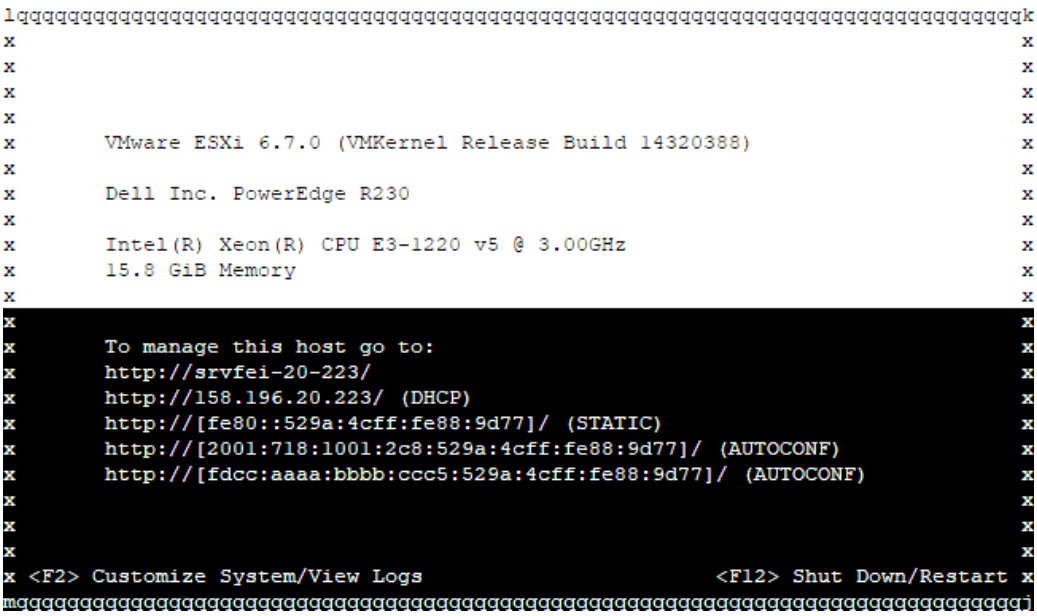
```
network:
    ethernets:
        nazovrozhrania:
            dhcp4: true
            nameservers: [158.196.0.53, 158.196.99.166]
    version: 2
    renderer: networkd
```

Následne som pomocou príkazu povolil uložené zmeny:

- `sudo netplan apply`

5.2 VMware ESXi 6.7.0

Inštalácia VMware ESXi prebehla bez problémov. Z oficiálnych stránok spoločnosti VMware bol stiahnutý inštalačný obraz vo formáte .iso, následne som si za pomoci programu Rufus[54] vytvoril bootovacie USB. Po načítaní USB serverom sa objavilo uvítacie okno, kde som potvrdil licenčné podmienky, vybral pevný disk na ktorý sa hypervizor nainštaluje, vybral rozloženie klávesnice a nakoniec zadal a potvrdil root heslo. Po týchto základných úkonoch sa spustila samotná inštalácia, ktorá zabrala niekoľko minút. Keď sa inštalácia dokončila, tak som bol vyzvaný vybrať bootovacie USB a reštartovať server.

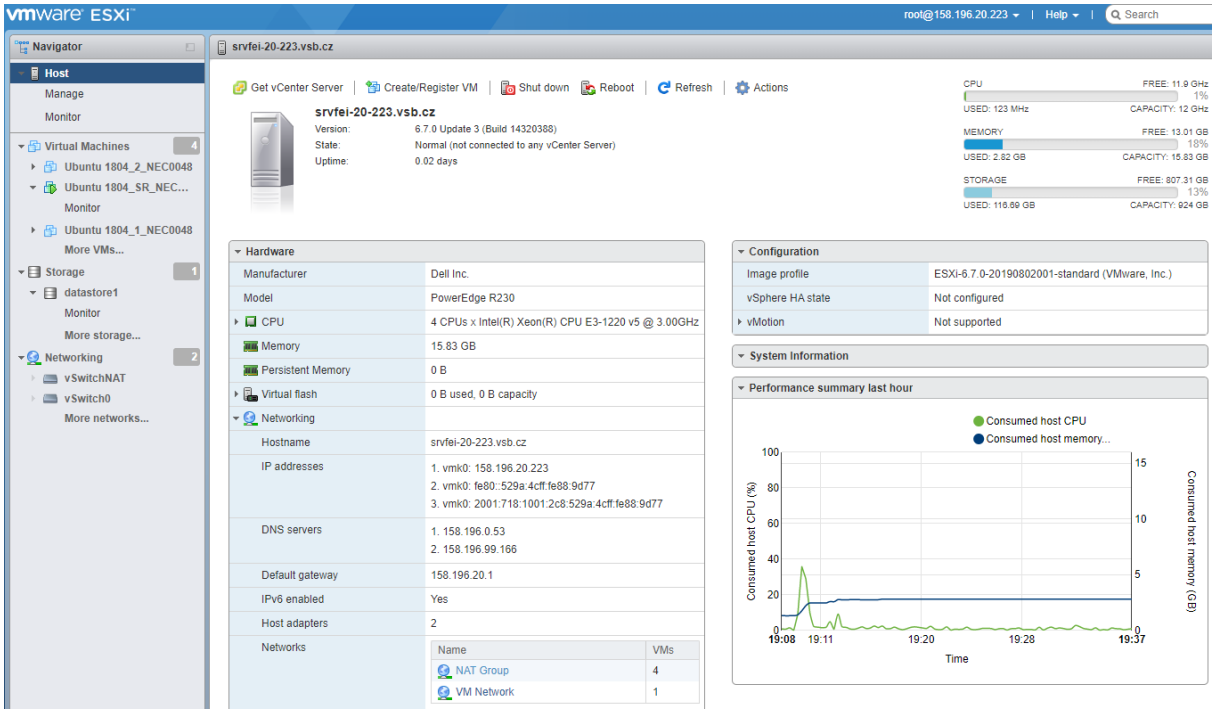


Obrázok 24: DCUI konzola hypervizora VMware ESXi

Ak všetko prebehlo v poriadku, tak sa na obrazovke zobrazila DCUI (Direct Console User Interface) konzola, ktorá slúži k základným informáciám alebo prípadnej prvotnej konfigurácií, túto konzolu môžeme jednoducho zobraziť aj v prípade vzdialeného pripojenia pomocou SSH, kde po pripojení stačí do príkazového riadku napísať príkaz **dcui**. Mimo iného tu nájdeme odkazy pomocou ktorých sa pripojíme na ESXi hostiteľa, pomocou klávesy F12 môžeme server reštartovať alebo vypnúť a klávesou F2 sa dostaneme do ponuky, kde môžeme konfigurovať základné veci okolo správy siete, zmeny root hesla, zobraziť záznamy a mnoho ďalšieho. Jedinou zmenou, ktorú som v DCUI konzoli vykonal, bolo povolenie SSH v „Customize System".

5.2.1 Vytvorenie virtuálnych strojov

Ku správe ESXi a jeho hostí som používal výhradne vstavaný webový klient ESXi Embedded Host Client.



Obrázok 25: Rozhranie vstavaného klienta ESXi

K pripojeniu som využíval aktualizovaný webový prehliadač Google Chrome a IP adresu verzie 4, podporované sú samozrejme aj IP adresy verzie 6. Po zadaní odkazu sa objavila uvítacia obrazovka, kde som vyplnil prihlasovacie údaje, zadané počas inštalácie a tým som sa pripojil k hostiteľovi.

Pred tým, ako som začal vytvárať jednotlivé virtuálne stroje som si najprv nahral Ubuntu Server 18.04.3 LTS vo formáte .iso do takzvaného „datastore" v záložke „Storage", ktorý je určený ako úložisko pre obrazy OS v .iso formáte, šablóny alebo súbory jednotlivých virtuálnych strojov. Takýchto úložísk si môžeme samozrejme vytvoriť viac. Po nahratí obrazu som sa mohol pustiť do vytvárania jednotlivých virtuálnych strojov.

Kliknutím na „Create/Register VM" sa objavil sprievodca vytvorením virtuálneho stroja, kde som zadal základné veci, ako názov, typ hosťovaného operačného systému, vybral úložisko „datastore" a pridelil virtuálny hardvér, ako CPU, operačná pamäť, pevný disk, sieťové adaptéry a podobne (virtuálny hardvér je možné meniť aj po vytvorení virtuálneho stroja). Ako posledné bolo potrebné zadať cestu k .iso obrazu operačného systému nahratého v „datastore". Tým som dokončil vytvorenie virtuálneho stroja, ktorý sa objavil v záložke „Virtual Machines". Akonáhle som spustil novovytvorený virtuálny stroj, tak som mohol začať inštaláciu OS tak, ako na klasický fyzický hardvér. V mojom prípade sa vždy jednalo o štandardnú inštaláciu spomenutého Ubuntu Server 18.04.3 LTS, ktorá bola rýchla a jednotlivé kroky myslím nebolo potrebné rozpisovať. Na záver som si skontroloval, či sa na každý virtuálny stroj automaticky nainštalovali VMware Tools, ktoré zvyšujú výkon a uľahčujú správu. Keby inštalácia neprebehla automaticky, ESXi hostiteľ na tento problém upozorní a odporučí inštaláciu jedným klikom.

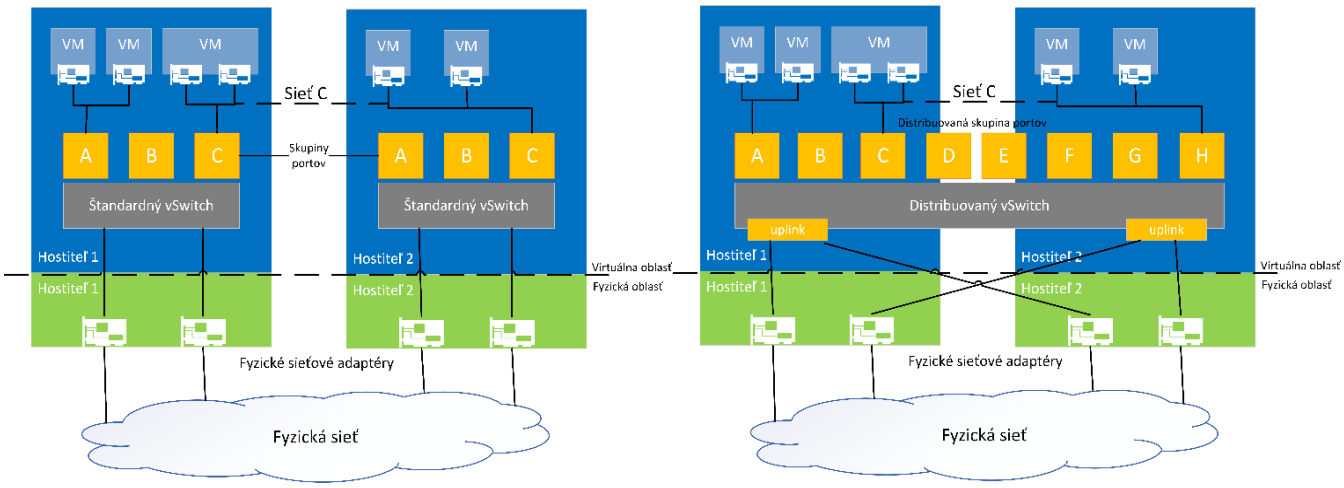
5.2.2 Možnosti konfigurácie siete a pripojenie virtuálnych strojov k internetu

Po vytvorení jednotlivých virtuálnych strojov bolo nutné nakonfigurovať sieť. VMware vSphere, ktorého je hypervizor ESXi súčasťou využíva takzvané virtuálne prepínače (vSwitches) zabezpečujúce sieťovú konektivitu. VMware vSphere používa dva typy virtuálnych prepínačov, štandardný (vSwitch) a distribuovaný (dvSwitch).

vNetwork Standard Switch (vSwitch) je virtuálny prepínač, ktorý poskytuje sieťovú konektivitu pre hostiteľa a jeho hostí. Tento virtuálny prepínač je možné použiť len v rámci jedného konkétneho ESXi hostiteľa. Je súčasťou voľne dostupného hypervizora ESXi.

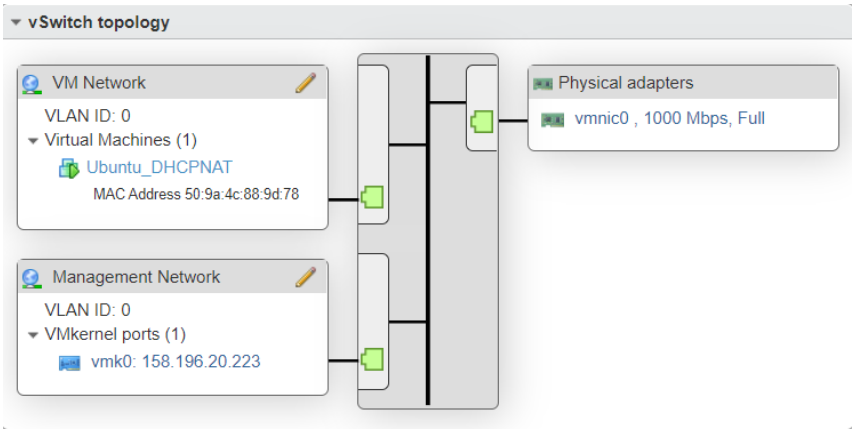
vNetwork Distributed Switch (dvSwitch) je virtuálny prepínač, ktorý umožňuje prepojiť viacero ESXi hostiteľov v rámci data centra. Poskytuje centralizované rozhranie pre konfiguráciu, monitorovanie a správu sieťovej infraštruktúry. Konfiguruje sa iba na vCenter servery[55]. Akonáhle je takýto prepínač nakonfigurovaný vo vCenter, tak všetky jeho nastavenia zdieľajú jednotlivý ESXi hostitelia, čo v prípade veľkej infraštruktúry uľahčuje a urýchluje prácu, pretože nemusíme na každom z nich konfigurovať samostatný štandardný vSwitch.

Ďalším dôležitým pojmom je skupina portov **Port Group** slúžiaca k oddeleniu skupín virtuálnych strojov na jednom virtuálnom prepínači.



Obrázok 26: Porovnanie vSwitch (vľavo) a dvSwitch (vpravo)

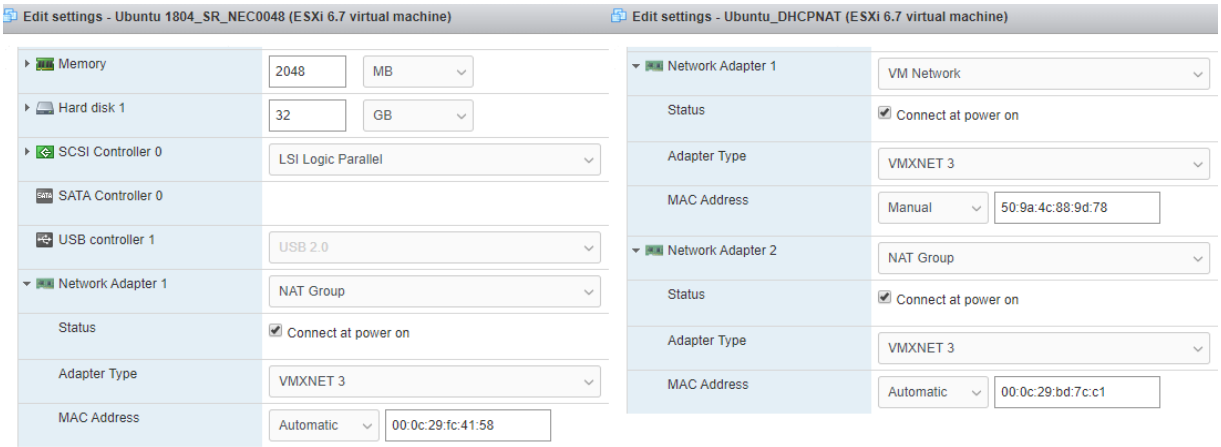
Po inštalácií má ESXi hostiteľ automaticky vytvorený štandardný virtuálny prepínač vSwitch0 s dvomi skupinami portov, VM Network a Management Network.



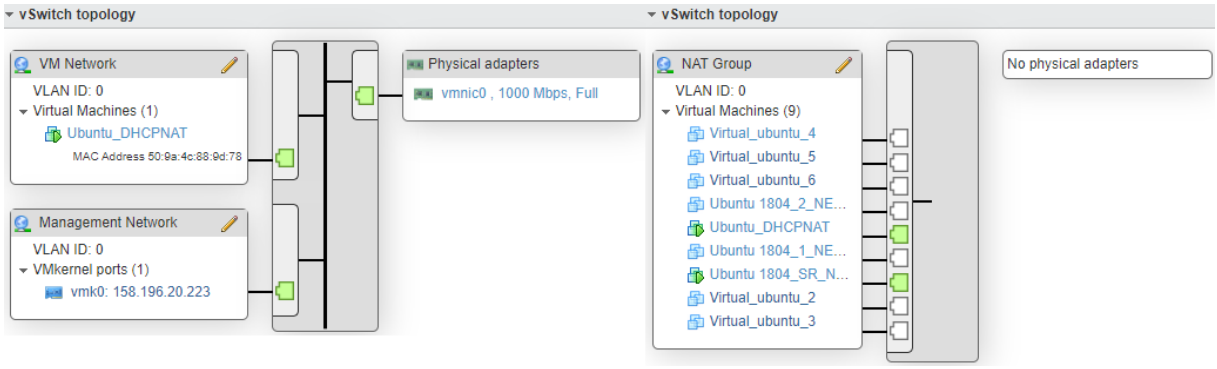
Obrázok 27: Automaticky vytvorený vSwitch0 hypervizorom ESXi

Ako môžeme vidieť na Obrázku 27, IP adresa na fyzickom rozhraní vmnic0 slúžila k pripojeniu a správe ESXi hostiteľa. Druhú, voľnú IP adresu na rozhraní vmnic1 som na základe MAC adresy pridelil virtuálnemu stroju Ubuntu_DHCPNAT. Ako som už spomenul virtuálny stroj Ubuntu_DHCPNAT slúžil ako základ virtuálnej siete, DHCP server a brána do internetu pre ďalšie virtuálne stroje. Pred samotnou konfiguráciou tejto brány som si oddelil svoju virtuálnu sieť a to tak, že som si vytvoril ďalší štandardný virtuálny prepínač a skupinu portov NAT Group. Ubuntu_DHCPNAT mal teda dva virtuálne sieťové adaptéry a jeho klienti len jeden. ESXi ponúka rôzne typy sieťových adapérov, ja som zvolil typ VMXNET3, medzi ďalšie ponúkané adaptéry patria:

- E1000 - emulovaná verzia gigabitového adaptéru Intel 82545EM.
- E1000e - emulovaná verzia gigabitového adaptéru Intel 82574.
- SR-IOV Passthrough - virtuálny stroj a fyzický sieťový adaptér si vymieňajú dáta bez VMkernelu ako sprostredkovateľa.
- VMXNET3 - paravirtualizovaný adaptér navrhnutý pre čo najvyšší výkon[56].



Obrázok 28: Konfigurácia sieťových adaptérov Ubuntu_DHCPNAT (vpravo) a klientov (vľavo)

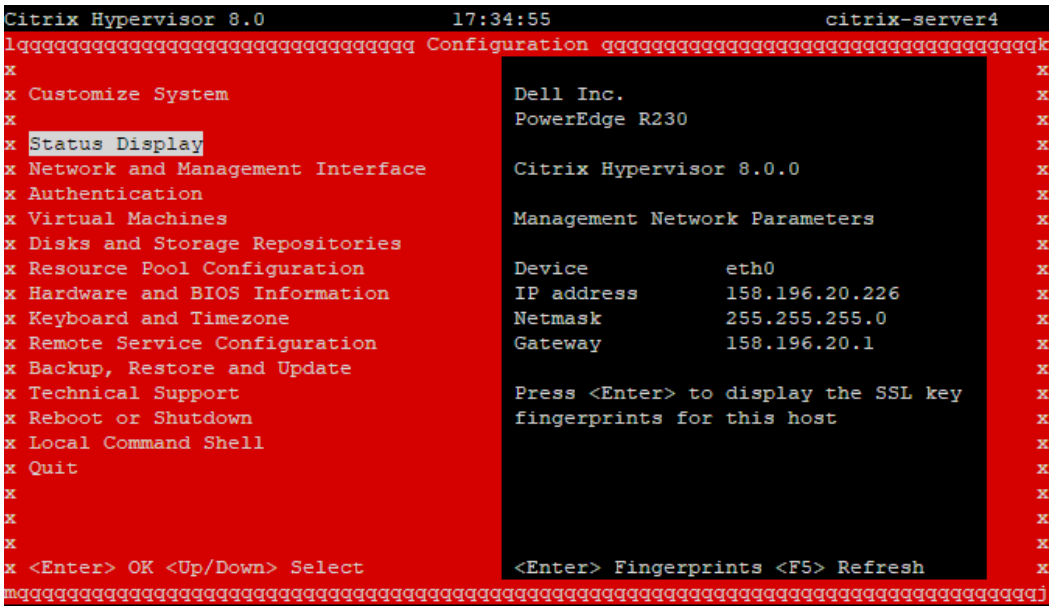


Obrázok 29: Topológia vytvorenej virtuálnej siete

Po vytvorení virtuálnej siete som mohol začať s konfiguráciou virtuálneho stroja Ubuntu_DHCPNAT. Ku konfigurácií som využil postup popísaný v podkapitole 5.1, s názvami rozhraní v podobe ens160 a ens192.

5.3 Citrix Hypervisor Express Edition 8.0

Inštalácia Citrix Hypervisora prebiehala podobne ako pri VMware ESXi. Z oficiálnych stránok spoločnosti Citrix bol stiahnutý obraz vo formáte .iso, nasledovalo vytvorenie bootovacieho USB. Po načítaní USB serverom nasledovalo potvrdenie základných parametrov, napríklad výber disku kam sa hypervizor nainštaluje, výber manažmentového rozhrania, zadanie hesla a podobne. Po nainštalovaní som bol vyzvaný k vybratiu USB a reštartu serveru. Ak všetko prebehlo v poriadku, tak sa zobrazila xsconsole konzola.



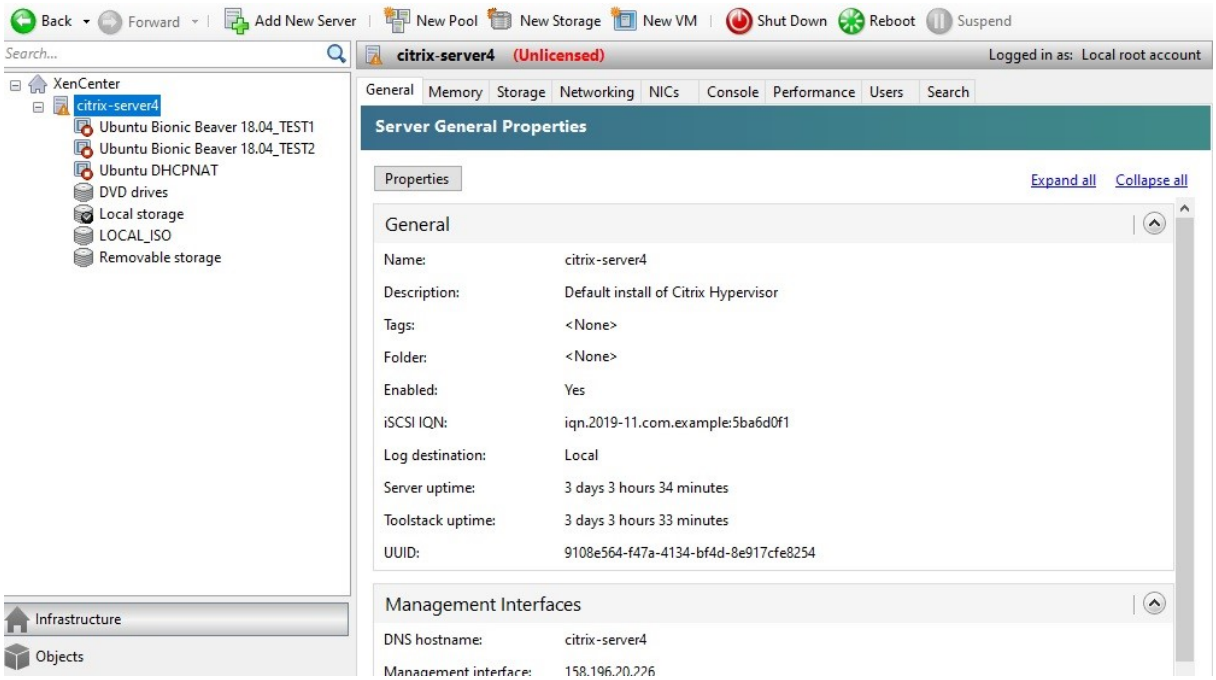
Obrázok 30: Rozhranie konzoly xsconsole Citrix Hypervisor

Ako môžeme vidieť na Obrázku 30, jej rozhranie je veľmi podobné DCUI VMware ESXi na Obrázku 24. Túto konzolu v prípade vzdialeného pripojenia na hostiteľa zobrazíme príkazom **xsconsole**.

Jedinou zmenou v konzoli bolo teda povolenie SSH pre vzdialené pripojenie v záložke „Remote Service Configuration".

5.3.1 Vytvorenie virtuálnych strojov

Ku správe hostiteľa som používal aplikáciu Citrix XenCenter, ktorej nespornou nevýhodou je dostupnosť len pre operačný systém Windows, takže tu sa Citrix a VMware podstatne líšia. Po stiahnutí a nainštalovaní Citrix XenCenter na svoj súkromný prenosný počítač som zadal prihlasovacie údaje, IP adresu a prihlásil sa na hostiteľa.



Obrázok 31: Rozhranie aplikácie Citrix XenCenter

Ako prvé som si vytvoril úložisko pre obrazy Ubuntu OS v .iso formáte, XenCenter taktiež ponúka pripojenie vlastného NFS alebo Microsoft SMB serveru, na ktorých máme uložené potrebné súbory. Ja som teda zvolil vytvorenie vlastného lokálneho úložiska s názvom LOCAL_ISO priamo na servery, ktoré som si vytvoril pomocou príkazu (záložka „Console"):

- `xe sr-create name-label=LOCAL_ISO type=iso device-config:location=/opt/xensource/packages/iso device-config:legacy_mode=true content-type=iso`

Po vytvorení som obraz Ubuntu Server 18.04.3 LTS nahral na server pomocou scp do umiestnenia, ktoré som zadal v príkaze pri vytváraní úložiska, teda **/opt/xensource/packages/iso**.

S nahratým obrazom Ubuntu Server 18.04.3 LTS som mohol začať s vytváraním virtuálnych strojov.

V záložke „VM” som zadal „New VM”, kde som ako prvé mal možnosť nainštalovať virtuálny stroj podľa rôznych šablón OS Windows a rôznych distribúcii OS Linux, ja som tieto šablóny nevyužil a pokračoval som v štandardnej inštalácií, kde som zadal názov, úložisko, špecifikoval cestu k .iso súboru OS a sieťové adaptéry. V XenCenter je možnosť nainštalovať virtuálny stroj so zavádzačom BIOS alebo UEFI, bezpečná cesta je mód BIOS, nakoľko UEFI je označené za experimentálnu funkciu, ktorá je dokonca vo väčšine prepripravených šablón automaticky zákázaná. Po nainštalovaní jednotlivých virtuálnych strojov som si na každý virtuálny stroj nainštaloval Citrix VM Tools, ktoré zastávajú podobnú funkciu ako VMware Tools.

5.3.2 Možnosti konfigurácie siete a pripojenie virtuálnych strojov k internetu

Sieť Citrix Hypervizora tvoria virtuálne prepínače. Pri vytváraní novej siete pomocou aplikácie XenCenter máme k dispozícií 4 typy sietí, každá z nich sa líší účelom použitia.

Single-Server Private Network je vnútorná sieť, ktorá nemá žiadnu spojitosť s fyzickým sieťovým adaptérom, poskytuje konektivitu len medzi virtuálnymi strojmi v rámci jedného konkrétneho hostiteľa, bez možnosti prístupu na internet.

Cross-Server Private Network je vnútorná sieť, ktorá poskytuje konektivitu pre virtuálne stroje v rámci serverovej oblasti, obsahujúcej viacero fyzických hostiteľov. Podobne ako v prípade Single-Server Private Network neposkytuje pre virtúalne stroje prístup na internet. Ku svojej činnosti vyžaduje inštaláciu a konfiguráciu Open vSwitch a vSwitch Controller.

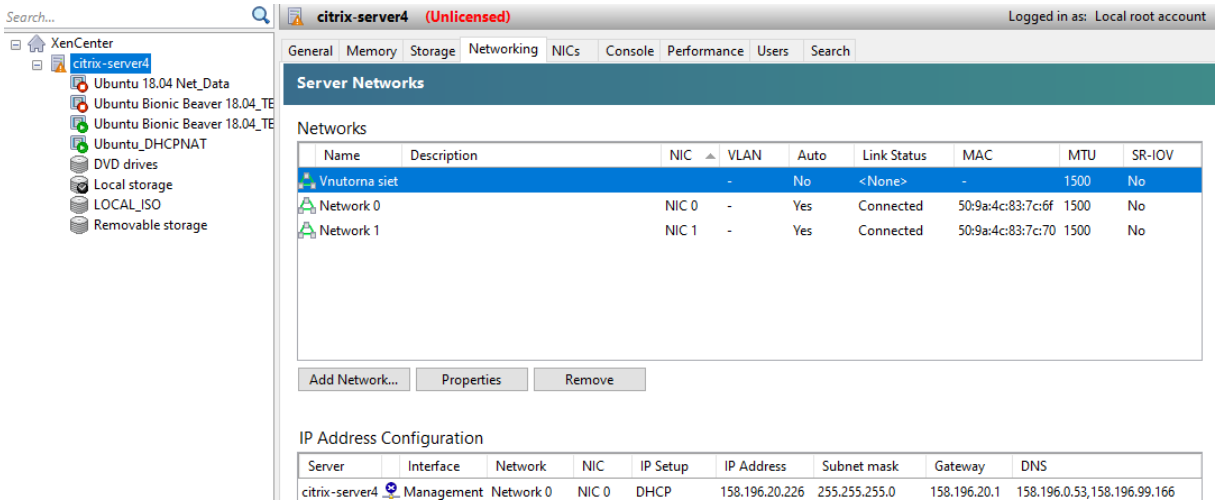
External Network je typ siete s prístupom k fyzickému sieťovému rozhraniu. Poskytuje most medzi virtuálnym strojom a fyzickým sieťovým rozhraním s prístpom na internet.

Bonded Network táto sieť je vytvorená spojením dvoch alebo viacero sieťových adaptérov, tým sa vytvorí jeden, vysoko výkonný kanál medzi virtuálnym strojom a sieťou[57].

Objekty siete sú dôležitým pojmom v prípade Citrix Hypervisora, reprezentujú 3 sieťové entity:

- **PIF**, reprezentuje fyzický sieťový adaptér hostiteľa.
- **VIF**, reprezentuje virtuálny sieťový adaptér virtuálneho stroja.
- **Network**, je vlastne virtuálny ethernetový prepínač hostiteľa, pozostáva z PIF a VIF.

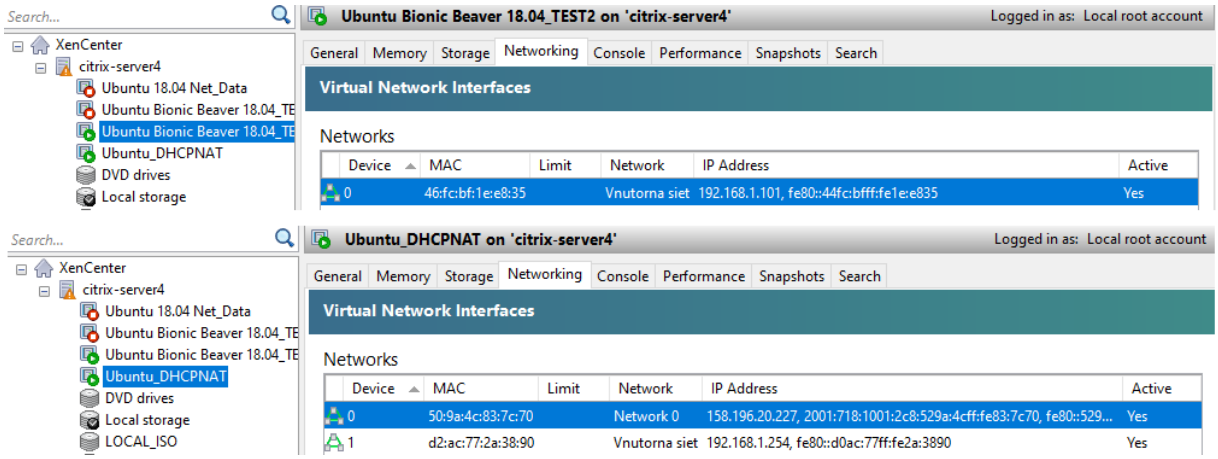
Po pripojení na hostiteľa pomocou XenCenter a kliknutí na záložku „Networking” môžeme vidieť aktuálny prehlaď vytvorených sietí.



Obrázok 32: Prehľad vytvorených sietí v aplikácii XenCenter

Siete Network 0 a Network 1 sú automaticky vytvorené externé siete (External Network), každá má pridelenú IP adresu na základe MAC adresy fyzického rozhrania. Ako môžeme vidieť na Obrázku 32, sieť Network 0 (IP Address Configuration) slúžila k pripojeniu a správe hostiteľa.

Ja som si mimo týchto dvoch automaticky vytvorených sietí vytvoril ďalšiu, súkromnú sieť v rámci jedného serveru (Single-Server Private Network) s názvom „Vnutorna sieť", ktorá slúžila ako základ mojej virtuálnej siete. Sieť Network 1 som pridelil virtuálnemu stroju Ubuntu_DHCPNAT, ktorý mal úplne totožnú funkciu ako v prípade VMware ESXi, teda brána do internetu pre virtuálne stroje vo vnútri virtuálnej siete.



Obrázok 33: Konfigurácia siete, brána Ubuntu_DHCPNAT (dole) a jej klientov (hore)

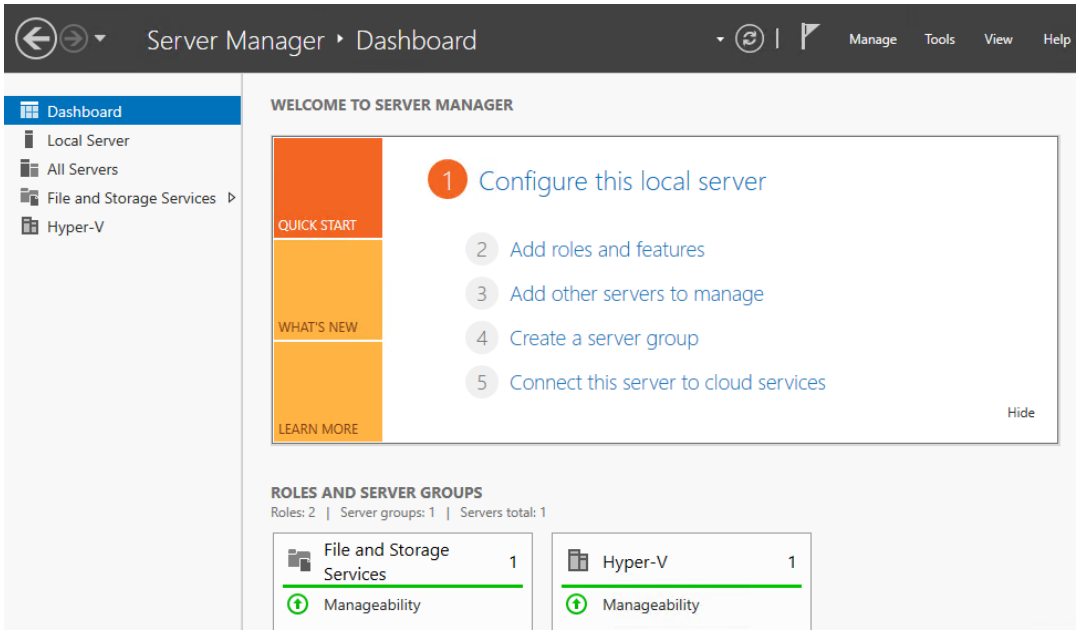
Ku konfigurácií brány Ubuntu_DHCP a jej klientov som využil úplne totožný postup ako pri VMware ESXi, teda podkapitola 5.1, jediným rozdielom boli názvy sieťových rozhraní. Citrix Hypervisor využíva názvy sieťových rozhraní v podobe eth, v mojom prípade eth0 a eth1 (VMware ESXi ens160 a ens192).

5.4 Microsoft Server Hyper-V 2019

Ďalšou virtualizačnou platformou je hypervizor od spoločnosti Microsoft, Hyper-V. Oproti dvom predchádzajúcim riešeniam prebiehala inštalácia odlišne. Hyper-V je hypervizor Typu 1, ale funguje na inom princípe ako Citrix Hypervisor a VMware ESXi. Ako som už spomenul v teoretickej časti, Hyper-V sa musí najskôr povoliť ako rola serverového operačného systému Windows, ktorého je súčasťou. Takže najprv som si z oficiálnych stránok Microsoft stiahol obraz .iso OS Windows Server 2019, ktorý je dostupný zdarma na testovanie po dobu 180 dní, pre účely mojej práce teda plne dostačujúce. Vytvoril som si bootovacie USB a následne začal s inštaláciou.

Nainštaloval som si Windows Server 2019 Standard s grafickým používateľským rozhraním (Desktop Experience). Po štandardnej inštalácií, veľmi podobnej inštalácií Windows 10 a reštartovaní serveru sa zobrazí grafické rozhranie, ktoré je nám známe práve z tohoto OS určeného pre klasické stolné počítače.

Ku správe využíva OS Windows Server takzvaný Server Manager, jedná sa o nástroj určený hlavne k spravovaniu jednotlivých rolí serveru.

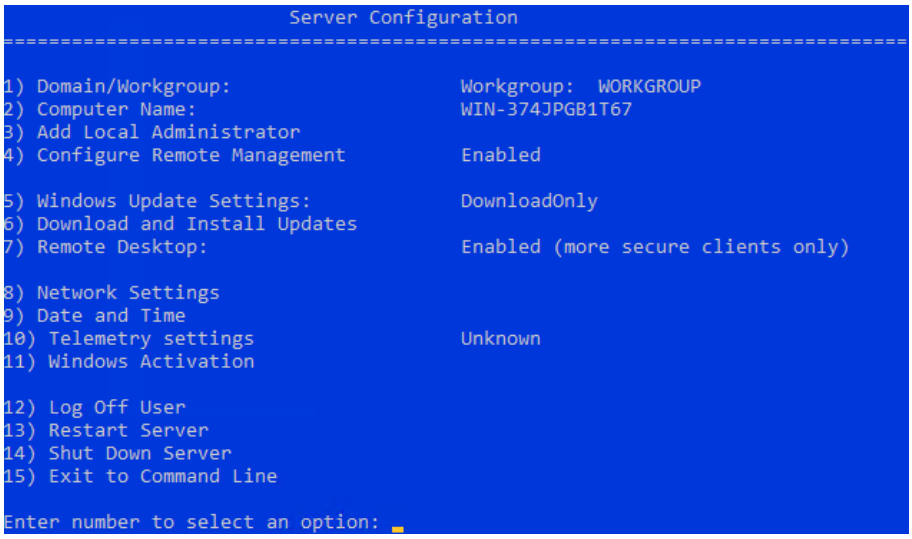


Obrázok 34: Microsoft Windows Server Manager

Na Obrázku 34 môžeme vidieť prostredie nástroja Server Manager, kde som v záložke „Add roles and features" pridal a nainštaloval rolu Hyper-V. Pre zaujímavosť tu nájdeme role, ako DHCP Server alebo DNS Server. Vo všeobecnosti spoločnosť Microsoft neodporúča na servery s už povolenou Hyper-V rolou inštalovať ďalšie role[58].

Keď sa táto rola úspešne nainštalovala, bol som vyzvaný reštartovať server. Po reštarte nastala v systéme podstatná zmena. Windows Server po nainštalovaní Hyper-V role funguje v podstate ako regulárny virtuálny stroj, určený ku správe Hyper-V infraštrukúry. Môžeme povedať, že sa zmenila jeho rola a je zodpovedný za vytváranie a správu podradených partícií, teda virtuálnych strojov. Táto hlavná partícia v podobe OS Windows Server je nazývaná ako koreňová partícia.

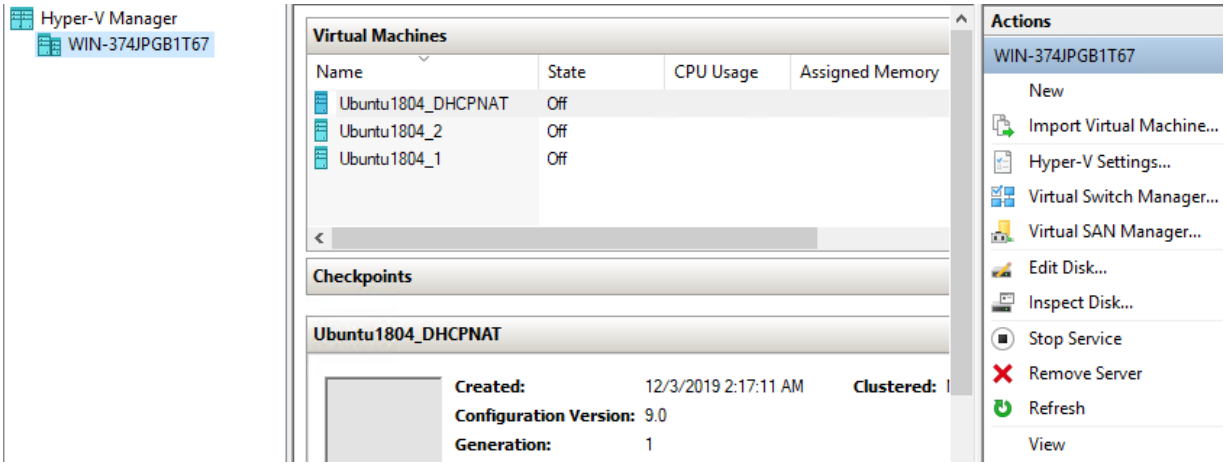
V prípade, že nemáme dostupné grafické rozhranie alebo je administrátorovi pohodlnejšie používanie príkazového riadku, tak môžeme server spravovať prostredníctvom konzole, ktorú zobrazíme príkazom **sconfig** do príkazového riadku cmd. Nevýhodou tejto konzole je, že neposkytuje správu rolí, ale je podobne, ako predchádzajúce riešenia určená skôr k prvotnej konfigurácii. Pre správu rolí potom slúžia samostatné príkazy.



Obrázok 35: Rozhranie konzoly Windows Server

5.4.1 Vytvorenie virtuálnych strojov

Správu hostiteľa a virtuálnych strojov mi zabezpečila aplikácia Hyper-V Manager, ktorá sa automaticky nainštalovala povolením Hyper-V role. Nakoľko som mal k dispozícii grafické rozhranie, tak som sa k serveru pripájal vždy prostredníctvom vzdialenej plochy pomocou RDP protokolu.



Obrázok 36: Rozhranie aplikácie Hyper-V Manager

Najskôr som si na hostiteľa prostredníctvom RDP nahral základ virtuálnych strojov, teda obraz OS Ubuntu Server 18.4.03 LTS. Samotné vytváranie jednotlivých virtuálnych strojov prebiehalo podobne ako pri predchádzajúcich platformách.

Po kliknutí na „New" a na „Virtual Machine" sa zobrazil sprievodca inštaláciou, takže som zadal základné veci ohľadom názvu, virtuálneho hardvéru, siete a cesty k obrazu .iso OS Ubuntu. Zaujímavou funkciou bola pri vytváraní voľba takzvanej generácie virtuálneho stroja.

- **Generácia 1**, 32 a 64 bitové verzie hosťovaných operačných systémov. Využíva zavádzač BIOS a podporuje širokú škálu OS.
- **Generácia 2**, využíva zavádzač UEFI a nové virtualizačné funkcie. V prípade hosťovaného OS Windows sa vyžaduje 64 bitová verzia, nepodporuje staršie OS, ako Windows 7 alebo Windows Server 2008, distribúcia Ubuntu je podporovaná od verzie 14.04[59].

Ja som pre svoje virtuálne stroje zvolil Generáciu 1. Po vytvorení a nainštalovaní virtuálnych strojov som začal s konfiguráciou siete.

5.4.2 Možnosti konfigurácie siete a pripojenie virtuálnych strojov k internetu

Najdôležitejším komponentom Microsoft Hyper-V siete je virtuálny prepínač, ktorý je možné nakonfigurovať v 3 rôznych módoch, súkromnom, vnútornom a verejnom móde.

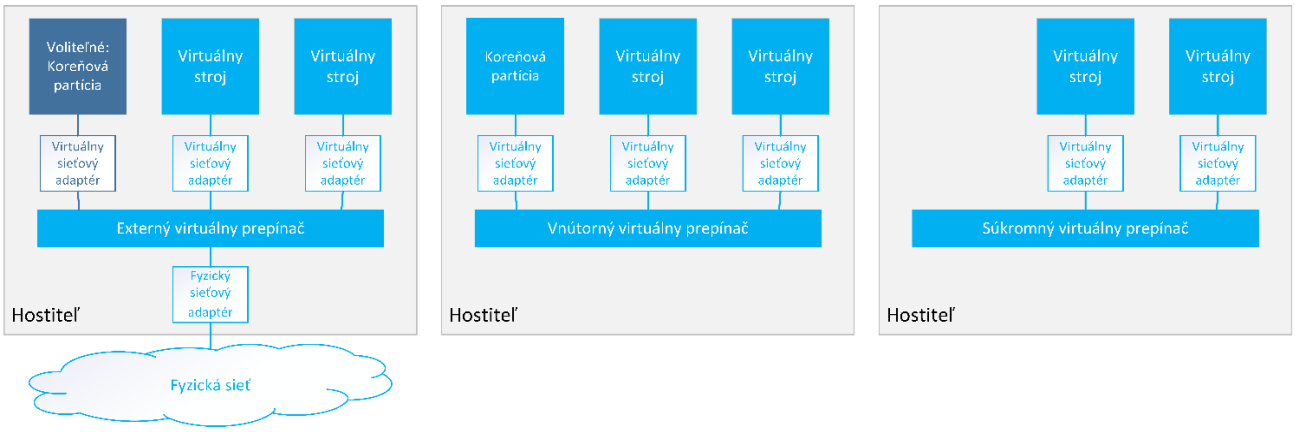
Hyper-V Internal Switch, poskytuje konektivitu medzi virtuálnymi strojmi a koreňovou partíciou[60]. Tento mód teda umožňuje koreňovej partícií priamo komunikovať s hociktorým virtuálnym strojom, ktorý má virtuálny adaptér na rovnakom vnútornom prepínači. Neposkytuje prístup k fyzickému sieťovému adaptéru.

Hyper-V Private Switch, je prepínač, ktorý je takmer identický ako vnútorný prepínač, ale s jediným rozdielom. Poskytuje konektivitu medzi virtuálnymi strojmi, ale neposkytuje prístup ku koreňovej partícií. Podobne ako vnútorný prepínač neposkytuje prístup k fyzickému sieťovému adaptéru.

Hyper-V External Switch, prepínač, ktorý musí byť pripojený k fyzickému sieťovému adaptéru s tým poskytuje prístup k internetu.

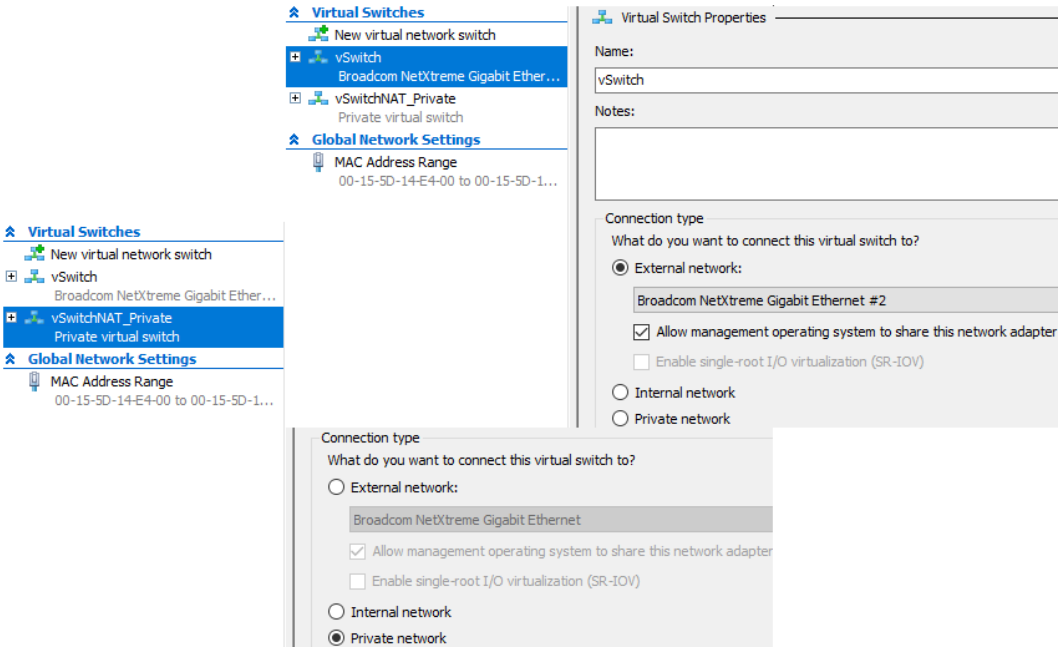
Tabuľka 4: Porovnanie Hyper-V prepínačov

	External Switch	Private Switch	Internal Switch
Prístup ku koreňovej partícií	Áno	Nie	Áno
Fyzický sieťový adaptér	Áno	Nie	Nie



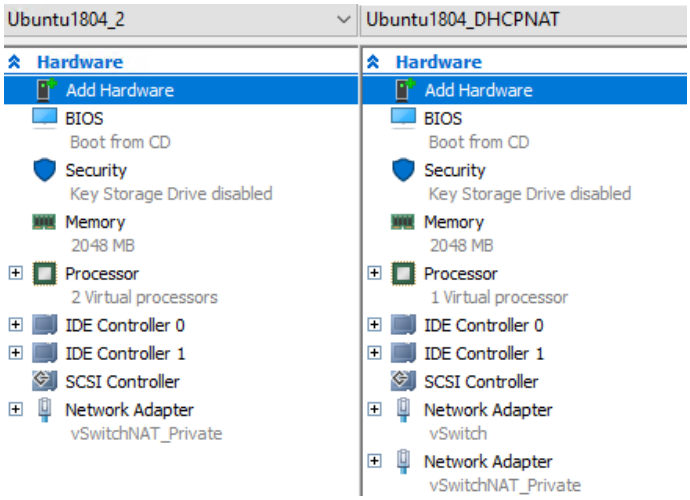
Obrázok 37: Porovnanie Hyper-V prepínačov

Základ mojej virtuálnej siete tvoril externý a súkromný virtuálny prepínač. IP adresa na rozhraní NIC 1 mi slúžila ku správe a pripojeniu na server. Druhú, voľnú IP adresu na rozhraní NIC 2 som pridelil vytvorenému externému virtuálnemu prepínaču (vSwitch), Obrázok 38.



Obrázok 38: Konfigurácia virtuálnych prepínačov Hyper-V

Podobne, ako pri predchádzajúcich dvoch virtualizačných platformách som si vytvoril virtuálny stroj Ubuntu_DHCPNAT, ktorý slúžil ako brána do internetu pre ostatných hostí. Tomuto virtuálnemu stroju som prostredníctvom Hyper-V Managera pridelil dva sieťové adaptéry, na jeden som pripojil externý virtuálny prepínač vSwitch s prístupom k internetu, druhý súkromný prepínač vSwitchNAT_Private mi slúžil k oblushe DHCP žiadostí. Klientom som teda vytvoril len jeden sieťový adaptér a pripojil súkromný prepínač.



Obrázok 39: Konfigurácia sieťových adaptérov Hyper-V

Akonáhle som si vytvoril a nakonfiguroval svoju virtuálnu sieť, mohol som pokračovať konfiguráciou brány Ubuntu_DHCPNAT a jej klientov. Microsoft Hyper-V využíva rovnaké názvy sieťových rozhraní ako Citrix Hypervisor, teda tvar v podobe eth, v mojom prípade eth0 a eth1. Takže ku konfigurácií brány som použil identický postup z podkapitoly 5.1, len som si musel dať pozor na názvy sieťových rozhraní.

5.5 KVM/QEMU

KVM je virtualizačná platforma, ktorá podporuje širokú škálu rôznych ditribúcií operačného systému Linux, ktorých je súčasťou. Menovať môžeme napríklad distribúcie ako CentOS, Fedora, openSUSE, Debian alebo Ubuntu. Ja som pre základ KVM zvolil distribúciu Ubuntu 18.04.3 LTS, pretože s touto distribúciou sa mi pracuje najlepšie.

Takže som si na server nainštaloval pomocou bootovacieho USB spomenutú distribúciu Ubuntu 18.04.3 LTS, zvolil som verziu s grafickým používateľským rozhraním GNOME. KVM vyžaduje ku svojej funkcií virtualizačné rozšírenia v podobe Intel VT alebo AMD-V. Pred samotnou inštaláciou KVM a jeho súčastí som si teda ešte overil, či server podporuje hardvérovú virtualizáciu, a či je v mojom prípade technológia Intel VT povolená v BIOSe serveru. K overeniu mi poslúžili nasledujúce príkazy:

- `egrep -c '(vmx|svm)' /proc/cpuinfo`

Ak je výstup vačší ako 0, tak systém podporuje hardvérovú virtualizáciu. V mojom prípade bol výstup 4.

- `sudo apt install cpu-checker`
- `sudo kvm-ok`

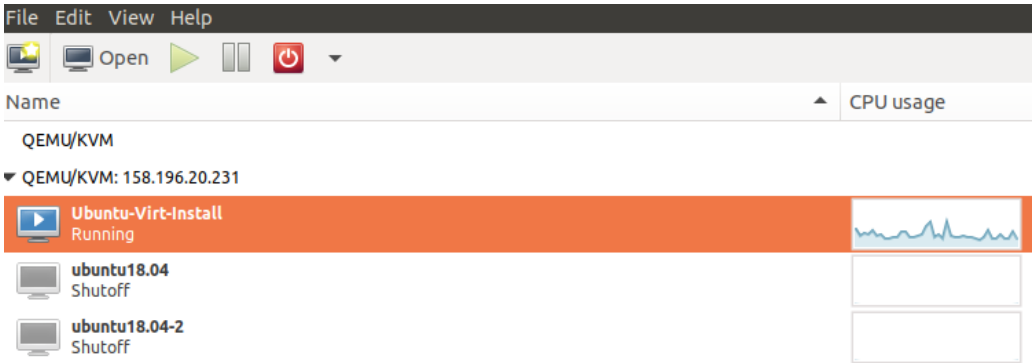
Príkazom kvm-ok som si overil, či je server schopný hosťovať hardvérovo akcelerované KVM virtuálne stroje. Výstup v podobe „**KVM acceleration can be used**" bola jasná odpoveď, na základe ktorej som mohol začať s inštaláciou samotného KVM a balíčkov, ktoré vyžaduje ku svojej funkcií a správe. K tomu mi poslúžili jednoduché príkazy:

- `sudo apt update`
- `sudo apt install qemu-kvm libvirt-bin bridge-utils virt-manager virt-viewer`

Týmto som mal všetko potrebné pripravené.

5.5.1 Vytvorenie virtuálnych strojov

Ku správe hostiteľa som využíval vzdialané pripojenie prosterdníctvom SSH, vzdialenú správu virtuálnych strojov a KVM infraštruktúry mi zabezpečila aplikácia virt-manager, ktorú som si nainštaloval na svoj súkromný počítač, na ktorom využívam dualboot OS Windows a OS Ubuntu. Sporadicky som využíval aj vzdialenú plochu prostredníctvom protokolu VNC.



Obrázok 40: Rozhranie aplikácie virt-manager

Na server som si pomocou scp prekopíroval z môjho počítača obraz OS Ubuntu a následne som začal s vytváraním a konfiguráciou virtuálnych strojov. Kliknutím na „File" a „New Virtual Machine" sa zobrazil sprievodca, kde som najskôr vybral spôsob inštalácie z lokálneho úložiska, na výber sú aj rôzne iné možnosti ako FTP alebo NFS. Nasledovala cesta k .iso súboru, typ a verzia OS a špecifikácia virtuálneho hardvéru.

Alternatíva pre skúsených administrátorov, ktorí preferujú príkazový riadok je virsh, respektíve virt-install. Virsh umožňuje správu KVM infraštruktúry, virt-install umožňuje vytváranie virtuálnych strojov. Ja som si takýto testovací vituálny stroj pomocou virt-install príkazu pre zaujímavosť vytvoril:

- `sudo virt-install --name=Ubuntu-Virt-Install \`

```
--vcpus=1 \
--ram=1024 \
--location 'http://archive.ubuntu.com/ubuntu/dists/bionic/main/
installer-amd64' \
--disk size=10 \
--os-type linux \
--os-variant ubuntu18.04 \
--network bridge:virbr0 \
--graphics none \
--extra-args "console=tty0 console=ttyS0,115200n8"
```

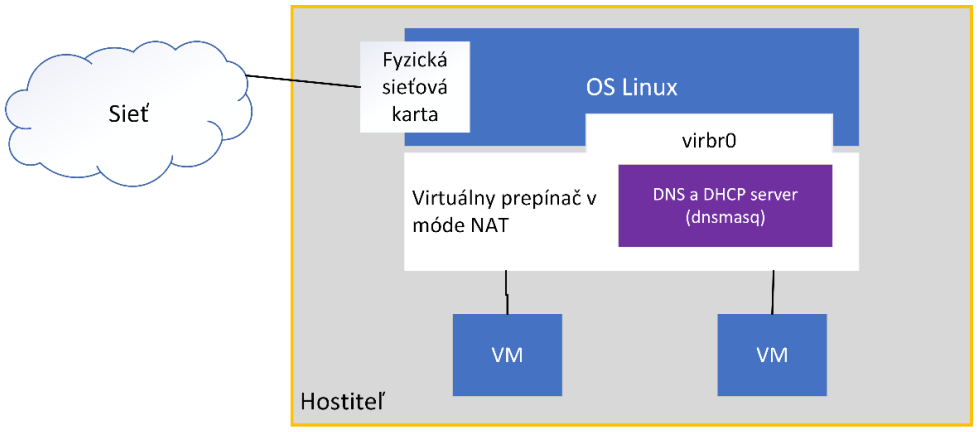
Takto vytvorený virtuálny stroj prostredníctvom virt-install sa mi dokonca hneď po vytvorení zobrazil v aplikácií virt-manager, Obrázok 40. Ako som už ale spomenul, pre správu virtuálnych strojov som výlučne používal virt-manager.

5.5.2 Možnosti konfigurácie siete a pripojenie virtuálnych strojov k internetu

KVM ponúka dve základné možnosti konfigurácie siete, takzvaný „Usermode Networking" a „Bridged Networking".

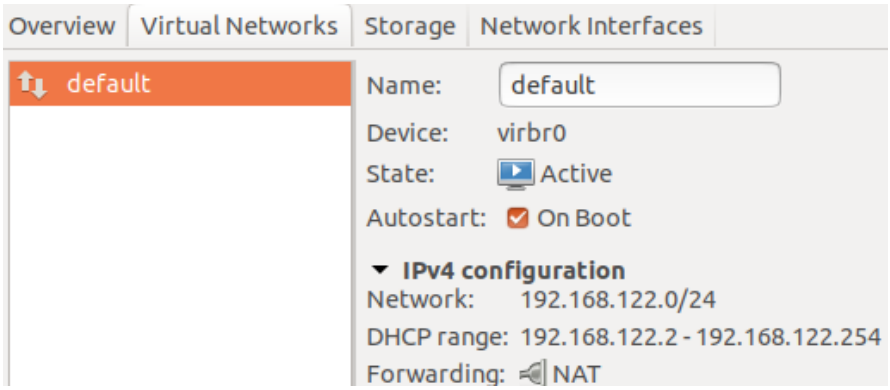
Usermode Networking, je štandardná predvolená konfigurácia, hosťovaný operačný systém má prístup k internetovým službám. Takýmto operačným systémom sú prideľované IP adresy vďaka vstavanému DHCP serveru, integrovanom priamo v QEMU. Predvolený adresný priestor, z ktorého sú adresy prideľované je 192.168.122.0/24, hostiteľský operačný systém je dostupný na adrese 192.168.122.1 [61]. Názov rozhrania, ktoré „Usermode Networking" využíva je **virbr0**, môžeme ho definovať ako virtuálny prepínač, ktorý je vytvorený automaticky inštaláciou služby libvirt a mimo integrovaného DHCP serveru, poskytuje aj DNS server a prístup do internetu prostredníctvom NAT. Veľkou výhodou je teda jeho automatická konfigurácia.

Bridged Networking, druhou možnosťou je most, ktorý umožňuje virtuálnemu sieťovému rozhraniu a tým viruálnemu stroju pripojenie k internetu cez fyzické sieťové rozhranie.



Obrázok 41: Usermode Networking

Pre virtuálnu sieť bol využitý predvolený „Usermode Networking" a automaticky vytvorená virtuálna sieť „default" s rozhraním virbr0. Tým som nemusel naviac inštalovať a konfigurovať virtuálny stroj a bránu do internetu Ubuntu_DHCPNAT, nakoľko jeho úlohu brány prebral základný operačný systém. Na Obrázku 42 môžeme vidieť sieť „default" a jej konfiguráciu.



Obrázok 42: Prehľad konfigurácie virtuálnej siete „default"

Jednotlivým virtuálnym strojom som následne už len pridelil virtuálnu sieť a tým som im umožnil prístup do internetu.

Obrázok 43: Pridelenie virtuálnej siete „default" virtuálnym strojom

Ako môžeme vidieť na Obrázku 46, mimo špecifikácie siete sa tu ešte nachádza model zariadenia. Ja som zvolil „virtio", jedná sa o virtualizačný štandard pre ovládače sieťových, prípadne diskových zariadení, kde ovládač zariadenia hosťa vie, že funguje vo virtuálnom prostredí a spolupracuje s hypervizorom. To umožňuje hosťovi zvýšiť výkon sieťových a diskových operácií[62]. Jedná sa teda o paravirtualizované ovládače.

5.6 LXD 3.0.3

Poslednou zo serverových virtualizačných platforiem je kontajnerový „lightervisor" LXD. Podobne ako pri predchádzajúcej virtualizačnej platforme KVM, som ako základ LXD použil Ubuntu 18.04.03 LTS. Na server som teda pomocou vytvoreného bootovacieho USB nainštaloval zvolenú distribúciu OS Linux. Ja som používal verziu LXD 3.0.3, jedná sa o dlhodobo podporovanú verziu, ktorá sa nachádza v repozitároch distribúcie Ubuntu 18.04. Veľkou výhodou LXD je jeho veľmi rýchla počiatočná konfigurácia v podobe pár jednoduchých príkazov.

Ako prvé som si aktualizoval repozitáre a nainštaloval démona LXD:

- `sudo apt update && sudo apt install lxd`

Po inštalácií vytvorí LXD skupinu používateľov s názvom lxd, kde je automaticky pridaný koreňový používateľ. Pri potrebe zabezpečiť oprávnenie pre iného ako koreňového používateľa stačí jeho používateľské meno pridať do skupiny lxd, v mojom prípade sa jednalo o používateľa nec0048:

- `sudo adduser nec0048 lxd`

K prvotnej a základnej konfigurácií využíva LXD príkaz lxd init, kde som si na základe série rôznych otázok nadefinoval sieť, úložisko a iné:

- `sudo lxd init`

```
ubuntulxd@ubuntulxd:~$ sudo lxd init
[sudo] password for ubuntulxd:
Would you like to use LXD clustering? (yes/no) [default=no]:
Do you want to configure a new storage pool? (yes/no) [default=yes]:
Name of the new storage pool [default=default]: BTRFS_storage
Name of the storage backend to use (btrfs, dir, lvm) [default=btrfs]: btrfs
Create a new BTRFS pool? (yes/no) [default=yes]:
Would you like to use an existing block device? (yes/no) [default=no]:
Size in GB of the new loop device (1GB minimum) [default=15GB]:
Would you like to connect to a MAAS server? (yes/no) [default=no]:
Would you like to create a new local network bridge? (yes/no) [default=yes]:
What should the new bridge be called? [default=lxdbr0]:
What IPv4 address should be used? (CIDR subnet notation, “auto” or “none”) [default=auto]:
What IPv6 address should be used? (CIDR subnet notation, “auto” or “none”) [default=auto]:
Would you like LXD to be available over the network? (yes/no) [default=no]:
Would you like stale cached images to be updated automatically? (yes/no) [default=yes]
Would you like a YAML "lxd init" preseed to be printed? (yes/no) [default=no]:
ubuntulxd@ubuntulxd:~$
```

Obrázok 44: Prvotná konfigurácia LXD

Ako môžeme vidieť na Obrázku 44, pri každej z otázok máme prednastavenú odpoveď. Ja som si nadefinoval nové úložisko, ktoré som si pomenoval BTRFS_storage a špecifikoval som súborový systém, v mojom prípade BTRFS, zvyšné otázky som potvrdil nezmenené. Po takejto základnej a rýchlej konfigurácií je LXD pripravené k činnosti.

5.6.1 Vytvorenie virtuálnych prostredí

Ku správe hostiteľa mi postačilo vzdialené pripojenie prostredníctvom SSH, nakoľko som ku správe využíval lxc klienta pracujúceho v príkazovom riadku. Pred samotným vytváraním a konfiguráciou virtuálnych prostredí som si najprv vytvoril ďalšie úložisko, ktoré ale využívalo iný súborový systém, ako to, ktoré som si nakonfiguroval v prvotnej konfigurácií. LXD ponúka aj možnosť súborového systému v podobe ZFS, takže som sa rozhodol vytvoriť aj tento typ úložiska.

V mojom prípade sa možnosť ZFS nenachádzala v ponuke, Obrázok 44, takže bolo nutné doinštalovať balíček zfsutils-linux, k tomu mi poslúžil jednoduchý príkaz:

- `sudo apt install zfsutils-linux`

Po nainštalovaní balíčka som znova zadal príkaz **lxd init** a podobne, ako pri prvotnej konfigurácií som si vytvoril druhé úložisko, tentokrát so súborovým systémom ZFS. Zobraziť jednotlivé úložiská môžeme príkazom:

- `lxc storage list`

```
ubuntulxd@ubuntulxd:~$ lxc storage list
+---------------+-------------+--------+--------------------------------------+---------+
|     NAME      | DESCRIPTION | DRIVER |                SOURCE                | USED BY |
+---------------+-------------+--------+--------------------------------------+---------+
| BTRFS_storage |             | btrfs  | /var/lib/lxd/disks/BTRFS_storage.img | 0       |
+---------------+-------------+--------+--------------------------------------+---------+
| ZFS_storage   |             | zfs    | /var/lib/lxd/disks/ZFS_storage.img   | 1       |
+---------------+-------------+--------+--------------------------------------+---------+
```

Obrázok 45: Vytvorené úložiská

S vytvorenými úložiskámi som začal s vytváraním jednotlivých virtuálnych prostredí. Kompletný výpis dostupných obrazov jednotlivých distribúcií OS Linux som si zobrazil príkazom:

- `lxc image list images:`

Dostupné obrazy konkrétnej distribúcie môžeme zobraziť napríklad:

- `lxc image list images: | grep -i "Ubuntu"`

Zaujímavosť, ktorú som už spomenul v teoretickej časti bola veľkosť obrazov, ktorá sa pohybovala v desiatkách megabajtov, veľkosť závisela od typu, verzie a použitej architektúry danej distribúcie.

Ja som si vytvoril virtuálne prostredia založené na Ubuntu 18.04.3 LTS, kde som špecifikoval verziu OS, názov a úložisko. **Ubuntu:** je názov prednastaveného vzdialeného úložiska obrazov OS Ubuntu. Konkrétne príkazy, ktoré som v práci pri vytváraní obrazov použil:

- `lxc launch ubuntu:18.04 Ubuntu-1804-BTRFS -s BTRFS_storage`
- `lxc launch ubuntu:18.04 Ubuntu-1804-ZFS -s ZFS_storage`

V prípade vytvorenia kontajnera založenom na Ubuntu alebo na inej distribúcií OS Linux môžeme použiť ďalšie prednastavené vzdialené úložisko **images:**. Odkaz na internetovú stránku, ktorá poskytuje zoznam všetkých dostupných obrazov úložiska images: môžeme najsť v použitej literatúre[63]. Ako príklad som uviedol príkaz pre vytvorenie kontajera založenom na Ubuntu 18.04.3 LTS a na distribúcií Fedora 31:

- `lxc launch images:ubuntu/bionic/amd64 Ubuntu-1804`
- `lxc launch images:fedora/31/amd64 Fedora-31`

Po vytvorení môžeme informácie o vytvorených kontajneroch zobraziť jednoduchým príkazom:

- `lxc list`

```
ubuntulxd@ubuntulxd:~$ lxc list
+------------------+---------+---------------------+-----------------------------------------------
+------------+-----------+
|       NAME       |  STATE  |        IPV4         |                     IPV6
|    TYPE    | SNAPSHOTS |
+------------------+---------+---------------------+-----------------------------------------------
+------------+-----------+
| Ubuntu-1804-BTRFS | RUNNING | 10.51.71.27 (eth0)  | fd42:51cc:1d6f:76b3:216:3eff:fecc:13f4 (eth0)
| PERSISTENT | 0         |
+------------------+---------+---------------------+-----------------------------------------------
+------------+-----------+
| Ubuntu-1804-ZFS   | RUNNING | 10.51.71.248 (eth0) | fd42:51cc:1d6f:76b3:216:3eff:fea6:74af (eth0)
| PERSISTENT | 0         |
+------------------+---------+---------------------+-----------------------------------------------
```

Obrázok 46: Výpis vytvorených kontajnerov

Akonáhle sa kontajner vytvoril, tak som sa doňho prihlásil napríklad príkazom:

- `lxc exec Ubuntu-1804-BTRFS -- /bin/bash`

Rozlíšiť to, či sa nachádzame v kontajneri môžeme jednoducho. V mojom prípade sa zmena prejavila tak, že sa príkazový riadok zmenil z ubuntulxd@ubuntulxd na root@Ubuntu-1804-BTRFS. Takto som sa uistil, že všetky zmeny, ktoré vykonávam sa dejú vo vnútri kontajneru. Napísaním príkazu **exit** som sa naopak vrátil späť do príkazového riadku hostiteľa.

Takže som si vytvoril jednotlivé virtuálne prostredia, ale zatiaľ som nikde nešpecifikoval hardvérové prostriedky. Po vytvorení prostredí som si všimol, ze každé prostredie má pridelené jedno jadro CPU a 2048 megabajtov pamäte RAM, k overeniu som využil príkaz htop. Predchádzajúce platformy ponúkali jednoduchú správu prostriedkov prostredníctvom grafického používateľského rozhrania. LXD k tomu využíva samozrejme prislúchajúce lxc príkazy. Výhodou je, že jednotlivé parametre je možné meniť priamo za chodu kontajera.

Ku správe CPU a jadier jednotlivých kontajnerov mi slúžil príkaz vo forme:

- `lxc config set nazov-kontajera limits.cpu X`

X znamená počet pridelených jadier procesora, ja som využíval 1 alebo 2. K dispozícií sú aj komplexné príkazy. Pokiaľ chceme prideliť napríklad konkrétne jadrá procesoru, povedzme druhé a štvrté jadro, tak môžeme využiť podobu limits.cpu 1,3.

Správu operačnej pamäte mi zabezpečil príkaz:

- `lxc config set nazov-kontajnera limits.memory X`

X znamená pridelenú veľkosť operačnej pamäte aj s jej jednotkou, príklad limits.memory 2048MB.

Nasledoval ďalší dôležitý príkaz, ktorý som využíval ku správe disku:

- `lxc config device set nazov-kontajera root size X`

X znamená veľkosť prideleného úložného priestoru aj s jednotkou, napríklad root size 5GB. K overeniu, či zmena prebehla úspešne mi slúžil príkaz df -h /, vykonaný vo vnútri kontajnera.

5.6.2 Možnosti konfigurácie siete a pripojenie virtuálnych prostredí k internetu

Pri prvotnej konfigurácií prostredníctvom lxd init som si zadefinoval nový sieťový most s predvolenými parametrami a názvom lxdbr0, Obrázok 44. Lxdbr0 zastáva podobnú funkciu virtuálneho prepínača, ako virbr0 v prípade „Usermode Networking" pri KVM. Pracuje v móde NAT a poskytuje vstavaný DHCP a DNS server[64]. V prípade, že nezadefinujeme vlastný adresný priestor, tak sú jednotlivým kontajnerom pridelované IP adresy z náhodného adresného priestoru, väčšinou sa jedná o tvar 10.X.X.X/24, Obrázok 46. Takto vytvorený most je automaticky pridelený všetkým novým kontajnerom a tým im poskytuje prístup do internetu. Lxdbr0 bol využitý ako základ mojej virtuálnej siete.

Takto automaticky vytvorená virtuálna sieť bola samozrejme plne dostačujúca, ja som sa ale rozhodol vytvoriť inú, testovaciu sieť využívajúcu adresný priestor 192.X.X.X/24. Táto sieť mi slúžila len na test alternatívnej tvorby siete a v práci nebola využívaná.

Začal som v podstate úplne od začiatku, ako prvé som úplne zmazal vytvorené kontajery a sieťový most lxdbr0:

- `lxc delete nazov-kontajera --force`
- `lxc network delete lxdbr0`

Vytvoril som si nový sieťový most s názvom testbr0:

- `lxc network create testbr0 ipv6.address=none ipv4.address=192.168.1.1/24 ipv4.nat=true`

Informácie o vytvorenom sieťovom moste som si zobrazil príkazom:

- `lxc network show testbr0`

Následne som si pomocou už známych príkazov vytvoril testovacie kontajnery test1 a test2 a priradil im vytvorený sieťový most testbr0:

- `lxc network attach testbr0 nazov-kontajera eth0`
- `lxc start nazov-kontajnera`

Po priradení siete testbr0 som pomocou príkazu lxc start spustil kontajery, pretože bezprostredne po vytvorení hlásili, že nie je k dispozícií žiadna sieť a automaticky sa prepli do vypnutého stavu. Príkazom lxc list som si overil pridelené IP adresy.

```
ubuntulxd@ubuntulxd:~$ lxc list
+-------+---------+----------------------+------+------------+-----------+
| NAME  |  STATE  |         IPV4         | IPV6 |    TYPE    | SNAPSHOTS |
+-------+---------+----------------------+------+------------+-----------+
| test1 | RUNNING | 192.168.1.214 (eth0) |      | PERSISTENT | 0         |
+-------+---------+----------------------+------+------------+-----------+
| test2 | RUNNING | 192.168.1.129 (eth0) |      | PERSISTENT | 0         |
+-------+---------+----------------------+------+------------+-----------+
```

Obrázok 47: Výpis vytvorených kontajnerov s alternatívnou sieťou testbr0

Ako môžeme vidieť na Obrázku 47, pridelené boli náhodné IP adresy, v prípade, že chceme kontajneru prideliť konkrétnu IP adresu, tak nám stačia prislúchajúce príkazy. Ja som sa rozhodol prideliť kontajneru test1 IP adresu 192.168.1.2. K tomu mi poslúžili príkazy:

- `lxc stop test1`
- `lxc config device set test1 eth0 ipv4.address 192.168.1.2`
- `lxc start test1`

Na záver by som ešte chcel dať do pozornosti jednu novú veľkú funkciu, ktorá sa prvýkrát objavila ako experimentálna v mesačnom vydaní LXD 3.19 vydaného 17.1.2020. Jedná sa o podporu klasických virtuálnych strojov[65].

6 Voľba virtualizačných platforiem pre operačné systémy Linux a Windows

Mimo piatich serverových virtualizačných platforiem som mal za úlohu otestovať aj tri platformy dostupné pre operačné systémy Linux a Windows. V tomto prípade hovoríme o riešeniach pre klasické stolné a prenosné počítače. Rozhodol som sa otestovať a porovnať dve riešenia využívajúce hypervizor Typu 2 a jedno riešenie využívajúce hypervizor Typu 1.

- **VMware Workstation 15.5.1 Player** - Hypervizor Typ 2.
- **Oracle VM VirtualBox 6.4.1** - Hypervizor Typ 2.
- **KVM/QEMU** - Hypervizor Typ 1.

6.1 Hardvérové prostriedky

Pre testovanie platforiem som využil svoj osobný prenosný počítač Acer Aspire A515-51G-55VR, na ktorom využívam „dualboot" OS Ubuntu Desktop 19.10 a Windows 10. Konfigurácia:

- **Procesor:** Intel Core i5-8250U / Frekvencia: 1.6 - 3.4GHz / Počet jadier: 4.
- **Operačná pamať:** 2x 4GB DDR4 2133MHz.
- **Pevný disk:** Micron 1100 MTFDDAV256TBN / Kapacita: 256GB / Windows 10.
  Western Digital WDS250G2B0A / Kapacita: 256GB / Ubuntu Desktop 19.10.
- **Sieťová karta:** RealTek RTL8168 PCI-E Gigabit Ethernet.

6.2 Operačný systém virtuálnych strojov

Podobne, ako pri serverovej virtualizácií bol pre virtuálne stroje použitý operačný systém Linux a to distribúcia **Ubuntu Server 18.04.3 LTS** „**Bionic Beaver**".

7 Inštalácia a konfigurácia virtualizačných platforiem pre Linux a Windows

V tejto kapitole som sa zaoberal inštaláciou a konfiguráciou zvolených virtualizačných platforiem pre operačné systémy Linux a Windows. Vďaka svojej dostupnosti a jednoduchosti použitia sú tieto platformy používané hlavne ako základ pre izolované testovacie prostredia rôznych aplikácií alebo operačných systémov. Nakoľko mám na svojom súkromnom prenosnom počítači izolovaný „dualboot" Windows 10 a Ubuntu 19.10, tak som sa rozhodol nainštalovať platformy VMware Workstation Player a Oracle VirtualBox na obidva spomenuté operačné systémy a sledovať nie len výkonové medzi platformami, ale aj rozdiely rovnakej platformy, ale pri rozdielnom hosititeľskom operačnom systéme. Tretia platforma QEMU/KVM bola otestovaná len na distribúcií Ubuntu Desktop 19.10, nakoľko je KVM neoddeliteľnou súčasťou Linux jadra.

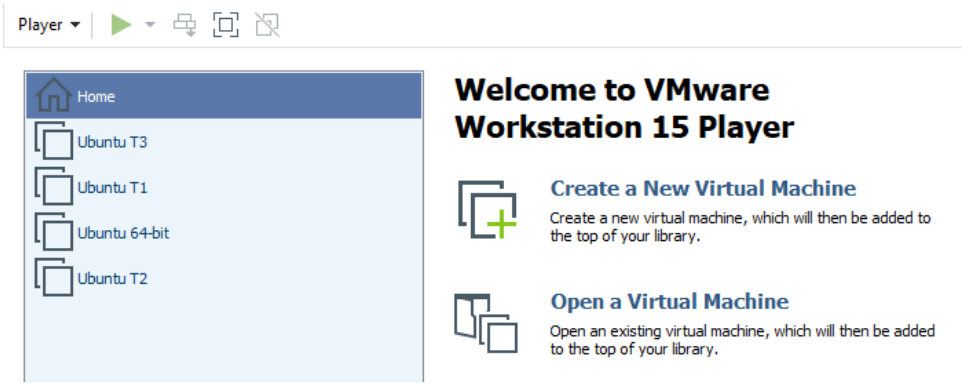
7.1 VMware Workstation 15.5.1 Player

Inštalácia na operačný systém Windows 10 prebiehala v pár jednoduchých krokoch, stačilo stiahnuť inštalačný súbor s príponou .exe z oficiálnych stránok spoločnosti VMware a nainštalovať pomocou prehľadného sprievodcu virtualizačnú platformu.

Inštalácia na operačný systém Ubuntu 19.10 mala podobný priebeh, ako pri Windows 10. Stiahol som si univerzálny inštalačný balíček pre OS Linux s príponou .bundle a pridal potrebné práva pre spustenie. K tomu som využil dva jednoduché príkazy:

- `sudo chmod +x VMware-Player-15.5.1-15018445.x86_64.bundle`
- `sudo ./VMware-Player-15.5.1-15018445.x86_64.bundle`

Ku správe využíva VMware Workstation Player prehľadné a moderné rozhranie.



Obrázok 48: Rozhranie VMware Workstation Player

Vytvorenie jednotlivých virtuálnych strojov mi zabralo niekoľko minút. Pre distribúciu Ubuntu Server 18.04.3 LTS je dostupná funkcia „Easy Install", ktorá po vytvorení virtuálneho stroja automaticky nainštaluje aj operačný systém. Postup vytvorenia virtuálneho stroja bol v zjednodušenej forme podobný platformám serverovej virtualizácie. Nasledovala konfigurácia siete, kde máme niekoľko možností.

VMware Wokstation Player ponúka nasledujúce typy sietí:

- **Bridged** (VMnet0) - Poskytuje priame pripojenie k fyzickej sieti.
- **NAT** (VMnet8) - Predvolená konfigurácia s prístupom k internetu pre všetky novovytvorené virtuálne stroje. IP adresa je pridelená na základe vstavaného DHCP serveru[66].
- **Host-Only** (VMnet1) - Súkromná sieť zdieľaná s hostiteľom, bez prístupu k internetu.
- **Custom** - Komplexné virtuálne siete, ktoré môžu byť pripojené k jednej alebo k viacero externým sieťam.

Ja som pre svoje virtuálne stroje využíval predvolenú sieť NAT.

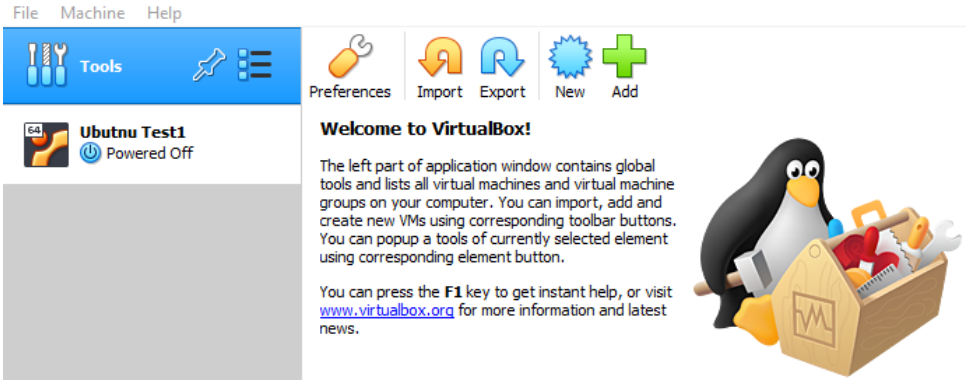
7.2 Oracle VM VirtualBox 6.1.4

Podobne, ako pri VMware Workstation Player bol z oficiálnych stránok stiahnutý inštalačný súbor pre operačný systém Windows, na základe ktorého som nainštaloval platformu.

Inštalácia v prípade operačného systému Ubuntu ponúka niekoľko možností. VirtualBox môžeme nainštalovať priamo z oficiálnych repozitárov, v ktorých sa ale nachádza staršia verzia. Ja som si nainštaloval najnovšiu verziu. Z oficiálnych stránok som si stiahol balíček .deb pre konkrétnu distribúciu, v mojom prípade Ubuntu 19.10 a pomocou príkazu gdebi nainštaloval aplikáciu:

- `sudo apt install gdebi-core`
- `sudo gdebi virtualbox-6.1_6.1.4-136177_Ubuntu_eoan_amd64.deb`

Ku správe využíva Oracle VM Virtualbox rozhranie, ktoré podľa môjho názoru pôsobí oproti VMware Workstation Player pomerne zastarane.



Obrázok 49: Rozhranie Oracle VM VirtualBox

Vytvorenie virtuálnych strojov zabralo podobne ako pri VMware Workstation Player niekoľko minút. Z môjho pohľadu bol ale sprievodca VirtualBox menej prehľadný v porovnaní s VMware Workstation Player a neponúka žiadnu podobnú funkciu „Easy Install". Ako jediná platforma počas vytvárania virtuálneho stroja nevyžadovala cestu k .iso súboru OS, tú som musel zadať v nastaveniach, až po vytvorení virtuálneho stroja. V prípade siete ponúka VirtualBox viacero možností konfigurácie, ako VMware Workstation.

Oracle VM Virtualbox ponúka nasledujúce typy sietí:

- **Internal** - Poskytuje konektivitu medzi virtuálnymi strojmi. Neposkytuje konektivitu s hostiteľom a externou sieťou.
- **NAT** - Poskytuje prístup do internetu pre virtuálne stroje vďaka NAT a vstavanému DHCP serveru. Nevýhodou je, že nie je umožnená vzájomná komunikácia medzi virtuálnymi strojmi.
- **NAT** N**etwork** - Rozširuje možnosti NAT o komunikáciu medzi virtuálnymi strojmi.
- **Bridged** - Priame pripojenie k fyzickej sieti.
- **Host-Only** - Zabezpečuje komunikáciu medzi virtuálnymi strojmi a hostiteľom, bez prístupu k fyzickej sieti[67].

Pre zabezpečenie konektivity som využíval sieť typu NAT Network.

7.3 KVM/QEMU

Ako poslednú virtualizačnú platformu pre klasické stolné a prenosné počítače som opäť zvolil KVM/QEMU. Výhodou tejto platformy je, že sa nejedná o čisto serverové riešenie, tak ako v prípade VMware ESXi alebo Citrix Hypervisor. Flexibilita tejto platformy spočíva v tom, že sa inštaluje na základný hostiteľský operačný systém Linux, ktorého kernel prekonvertuje do hypervizora Typu 1 vďaka KVM modulu. V kombinácií s emulátorom QEMU vzniká výkonná virtualizačná platforma.

Ja som si teda na svoj osobný prenosný počítač s operačným systémom Ubuntu Desktop 19.10 nainštaloval KVM/QEMU úplne identickým spôsobom, ako pri serverovej virtualizácií, popísanej v kapitole 5.5. Správu infraštruktúry mi zabezpečil virt-manager a ku konfigurácií siete bol opäť využitý „Usermode Networking".

8 Metodika testovania a porovnanie serverových virtualizačných platforiem

Cieľom jednotlivých testov bolo porovnanie všetkých piatich platforiem. Pri testovaní som sa zameral na výkon procesoru, prenosovú rýchlosť operačnej pamäte a jej správu, výkon súborového systému a virtuálnej siete. Všetky testy boli realizované na operačnom systéme Ubuntu Server 18.04.3 LTS. Pre referenciu bola pri niektorých testoch zaradená aj fyzická inštalácia operačného systému Ubuntu Server 18.04.3 LTS.

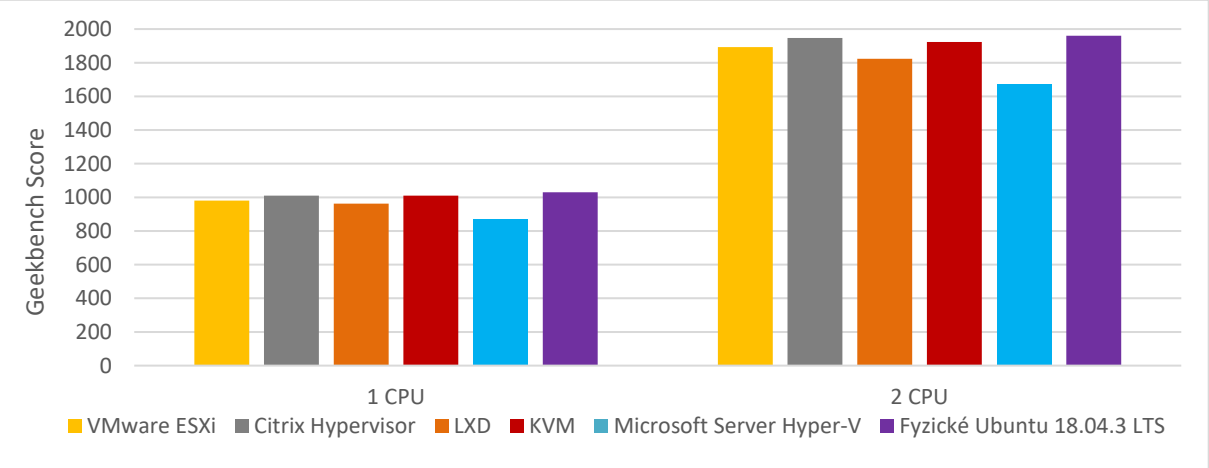
8.1 Test výkonu procesoru

Ako prvé som sa zameral na testovanie výkonu procesoru. Konfigurácia serverov, ktoré som mal k dispozícií obsahovala procesor Intel Xeon E3-1220 v5. Jedná sa o štvorjadrový a štvorvláknový procesor bez podpory technológie „Hyper-threading". K testovaniu som zvolil tri porovnávacie nástroje Geekbench 5, Sysbench a Phoronix Test Suite - Timed Linux Kernel Compilation.

Pre nástroje Geekbench 5 a Sysbench som vytvoril dve scenáre. Prvý scenár obsahoval virtuálny stroj s prideleným jedným alebo dvomi CPU, ostatné virtuálne stroje boli počas testu vypnuté. Druhý scenár bol totožný s prvým, ale tu som pred testom najskôr pomocou aplikácie stress maximálne vyťažil procesor serveru prostredníctvom ďalšieho virtuálneho stroja s pridelenými štyrmi CPU. Posledný test Timed Linux Kernel Compilation prebiehal vždy na virtuálnom stroji s pridelenými štyrmi CPU a vypnutými ostatnými virtuálnymi strojmi a to z dôvodu jeho celkovej naročnosti. Každý test bol spustený trikrát, výsledná a použitá hodnota bol aritmetický priemer nameraných výsledkov.

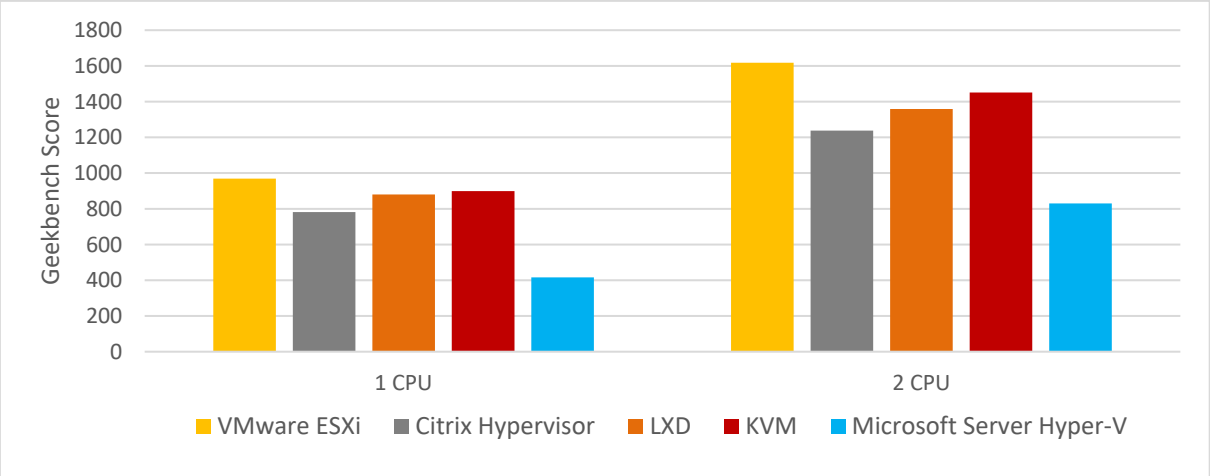
8.1.1 Geekbench 5

Základný porovnávací nástroj bol zvolený multiplatformový Geekbench 5, ktorý je zameraný na jedno a viacjadrový výkon procesoru. Obsahuje testy zamerané napríklad na šifrovanie, kompresiu alebo celočíselné matematické operácie. Výsledkom je takzvané „Geekbench Score", ktoré je kalibrované voči zakladnému skóre 1000 procesoru Intel Core i3-8100[68]. Na Grafe 1 môžeme vidieť, že ani jedna platforma výrazne nezaostáva, mierny prepad môžeme pozorovať u Hyper-V.



Graf 1: Geekbench 5 CPU Score bez zaťaženia serveru

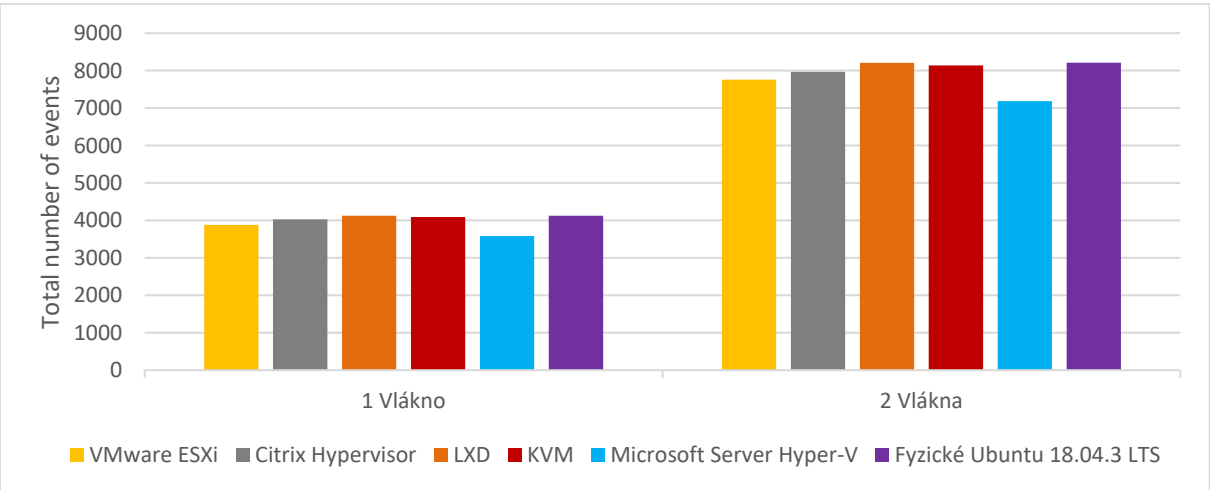
Zaujímavejšie výsledky priniesol druhý scenár s maximálne vyťaženým procesorom prostredníctvom druhého virtuálneho stroja a následnými testami. Ako môžeme vidieť na Grafe 2, veľký prepad o viac ako 50% zaznamenal Microsoft Server Hyper-V. Naopak, výborne výsledky v tomto teste preukázal VMware ESXi.



Graf 2: Geekbench 5 CPU Score so zaťažením serveru

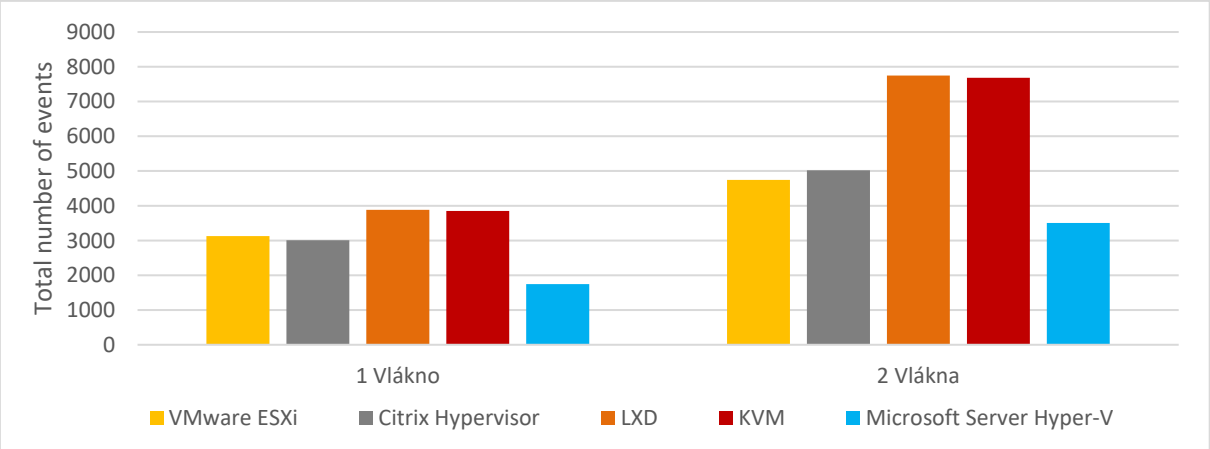
8.1.2 Sysbench CPU

Sysbench je porovnávací nástroj určený pre jedno alebo viacvláknové testovanie CPU. Mimo testovania CPU podporuje aj testovanie operačnej pamäte, súborového systému alebo MySQL databáz. V režime CPU každá požiadavka spočíva vo výpočte provčísel, až do stanovenej hodnoty[69]. Výsledkom je počet udalostí za určitú dobu, ktorú v príkaze stanovil používateľ. Ako môžeme vidieť na Grafe 3, LXD a KVM dosiahli takmer identický výsledok, ako fyzická inštalácia. Hyper-V zaznamenalo opäť mierny prepad v porovnaní s ostatnými platformami.



Graf 3: Sysbench CPU bez zaťaženia serveru

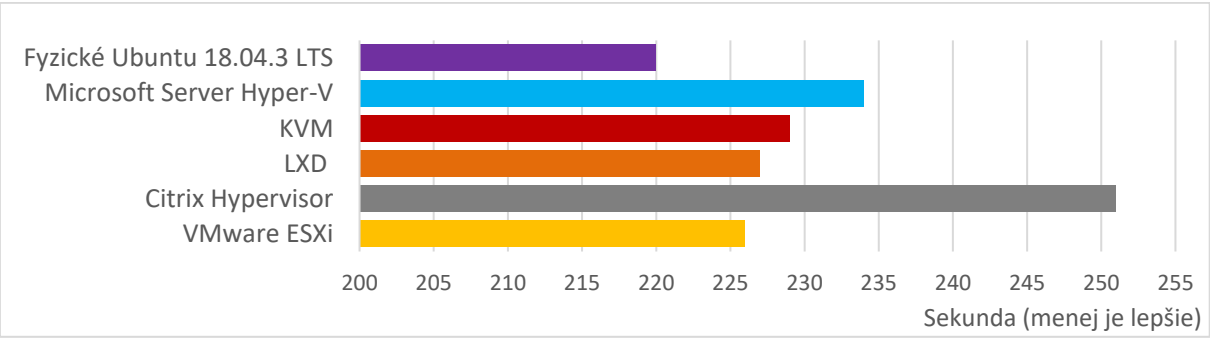
Graf 4 ukazuje výsledky so zaťaženým serverom. Zaujímavosťou je, aké vyrovnané výsledky podávali v tomto teste LXD a KVM, ktoré dosiahli výsledky priemerne len o 5% nižšie v porovnaní s výsledkami z Grafu 3 bez zaťaženia serveru. Približný 50% prepad zaznamenalo Hyper-V. Môj osobný pohľad na celkové výsledky som zhrnul na konci tejto kapitoly.



Graf 4: Sysbench CPU so zaťažením serveru

8.1.3 Timed Linux Kernel Compilation

Posledným testom bol Timed Linux Kernel Compilation[70], ktorý je súčasťou platformy Phoronix Test Suite[71]. Tento test je zameraný na dobu potrebnú na zostavenie Linux jadra v štandardnej konfigurácií. Výsledkom je čas v sekundách. Ako môžeme vidieť na Grafe 5, najlepšie výsledky podávali VMware ESXi a LXD, ktorým zostavenie jadra trvalo približne o 6, respektíve 7 sekúnd viac, ako fyzickej inštalácií. Naopak veľké problémy mal v tomto teste prekvapujúco Citrix Hypervisor, ktorý zaostal za Hyper-V o 17 sekúnd, voči fyzickej inštalácií dokonca o 31 sekúnd.



Graf 5: Timed Linux Kernel Compilation

8.1.4 Zhrnutie

Podľa vykonaných testov je zrejmé, že výkon procesoru v prípade nezaťaženého procesoru nie je zásadne negatívne ovplyvnený virtualizáciou. V prípade plného zaťaženia procesoru si myslím, že hrá hlavnú plánovač, prípadne jeho typ pri každej z platforiem, ktorý má na starosti rozdeľenie prostriedkov medzi jednotlivé virtuálne stroje.

8.2 Test výkonu operačnej pamäte

Pre test operačnej pamäte boli zvolené dva porovnávacie nástroje RAMspeed SMP[72] a komplexný Sysbench[69].

Pre porovnávací nástroj RAMspeed SMP som zvolil dva scenáre podobné tým, aké boli použité pri testovaní výkonu procesoru. Prvý, základný test prebiehal len s jedným spusteným virtuálnym strojom s pridelenými dvomi CPU a 2048MB RAM, na ktorom prebiehal test. Pri druhom scenári som prostredníctvom druhého virtuálneho stroja s pridelenými štyrmi CPU a nástroja stress maximálne vyťažil procesor serveru a následne opať spustil test.

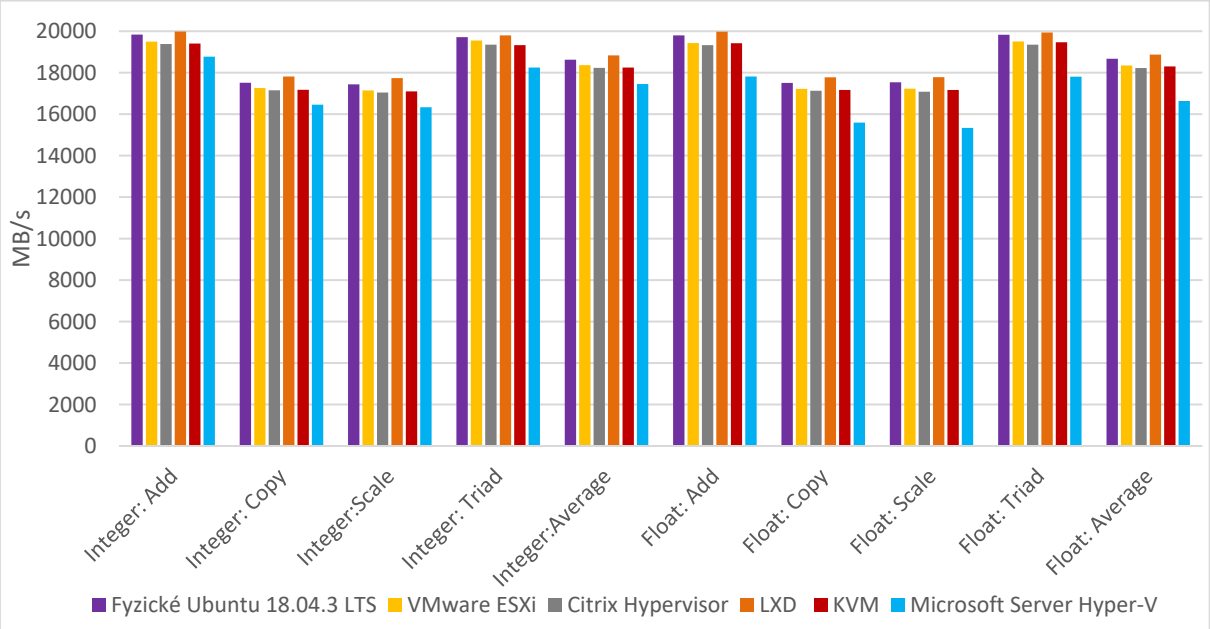
Pre Sysbench som zvolil mierne odlišný prístup. Opäť boli zvolené dva scenáre, ale tentokrát som sa zameral na vplyv veľkosti vyrovnávacej pamäte. Virtuálne stroje mali počas tohoto testu pridelené jedno alebo dve CPU a 2048MB RAM, ostatné virtuálne stroje boli vždy vypnuté. Jediný parameter, ktorý som menil bola alokácia vyrovnávacej pamäte, konkrétne 1kB a 1MB.

Každý z testov bol spustený trikrát, výsledná a použitá hodnota bol artimetický priemer nameraných výsledkov.

Po jednotlivých testoch som sa ešte zameral na využitie operačnej pamäte novovytvorených virtuálnych strojov/prostredí a na dynamickú správu operačnej pamäte z pohľadu hostiteľa.

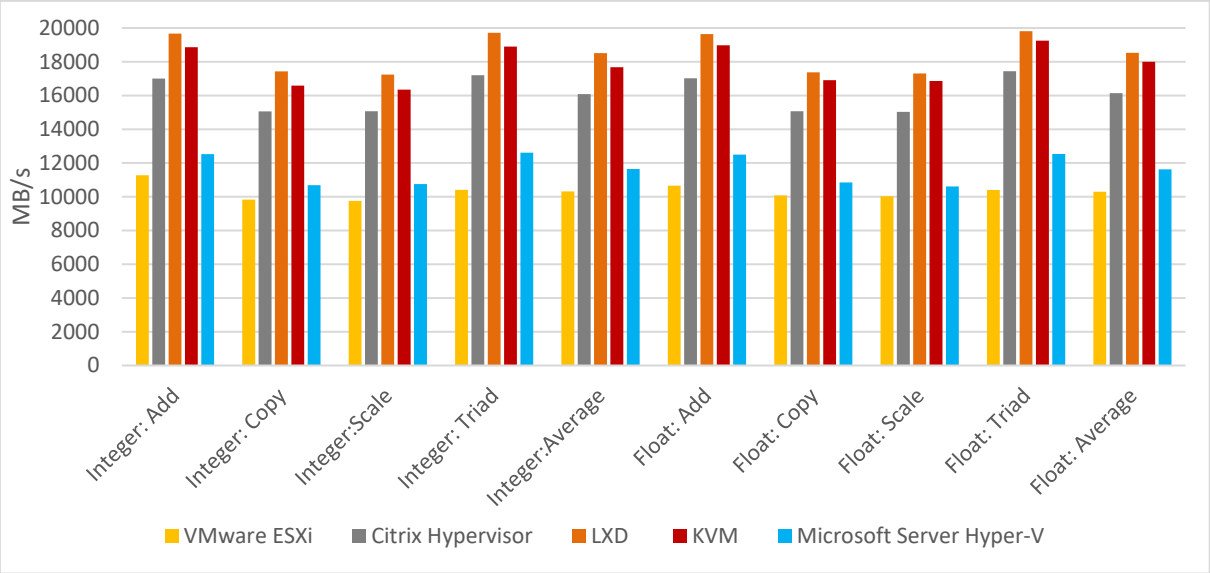
8.2.1 RAMspeed SMP

RAMspeed SMP je súčasťou platformy Phoronix Test Suite, ktorý je zameraný na testovanie výkonu operačnej pamäte. Ako môžeme vidieť na Grafe 6, platforma LXD dosiahla výborné výsledky, ktoré boli dokonca v priemere o 1.5% lepšie, ako v prípade fyzickej inštalácie. Mierne vyšší výkonnostný rozdiel zaznamenalo Hyper-V, ktoré zaostalo hlavne v operáciach čísel s pohyblivou rádovou čiarkou oproti fyzickej inštalácií priemerne o 13%.



Graf 6: RAMspeed SMP bez zaťaženia serveru

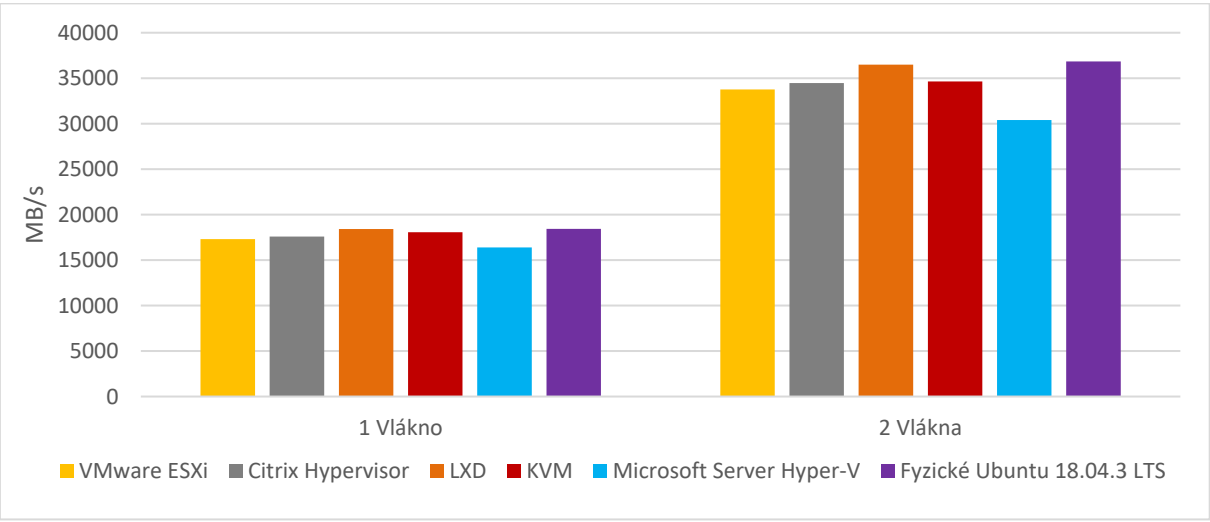
Graf 7 zobrazuje výsledky pri zaťaženom serveri. Pri tomto teste môžeme vidieť znateľné rozdiely medzi platformami. LXD zaznemenal výsledky približne o 1.5% nižšie oproti výsledkom z Grafu 6, zaujímavé výsledky podala aj platforma KVM. Najväčšie problémy mali v tomto teste VMware ESXi a Microsoft Hyper-V, ktoré zaznamenali prepad až o 44%, respektíve 34% oproti výsledkom z Grafu 6.



Graf 7: RAMspeed so zaťažením serveru

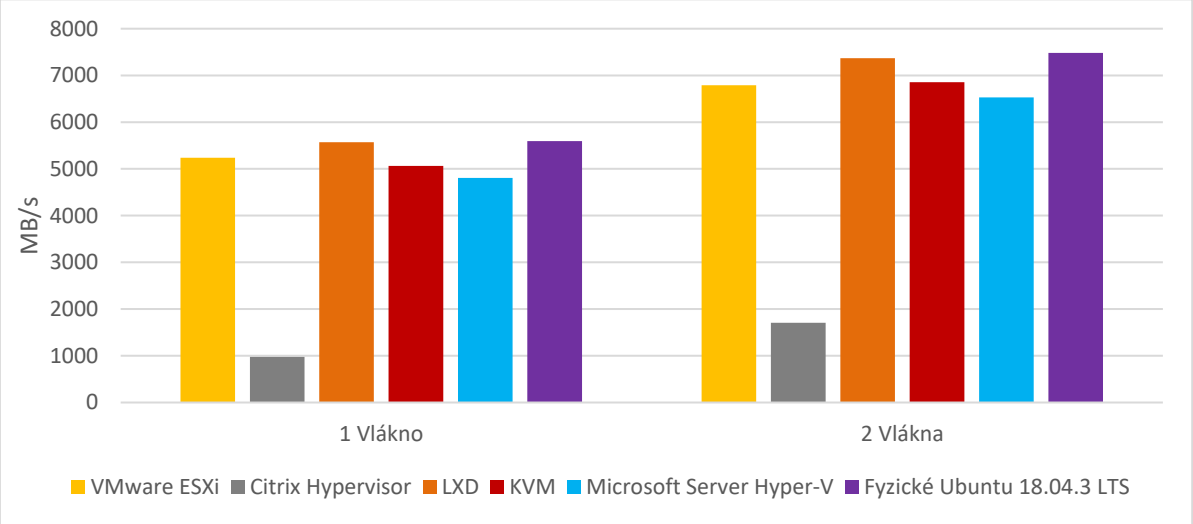
8.2.2 Sysbench RAM

V prvom teste operačnej pamäte som alokoval v nástroji Sysbench veľkosť vyrovnávacej pamäte prostredníctvom parametra na 1MB a celkovú veľkost dát určených k prenosu na 10GB. Ako možeme vidieť na Grafe 8, výsledky jednotlivých platforiem sa oproti fyzickej inštalácií nijak zvlášť nelíšili, výnimku tvoril v teste s dvoma vláknami Microsoft Hyper-V, ktorý zaostal voči fyzickej inštalácií približne o 17%.



Graf 8: Sysbench RAM, 1MB vyrovnávacia pamäť

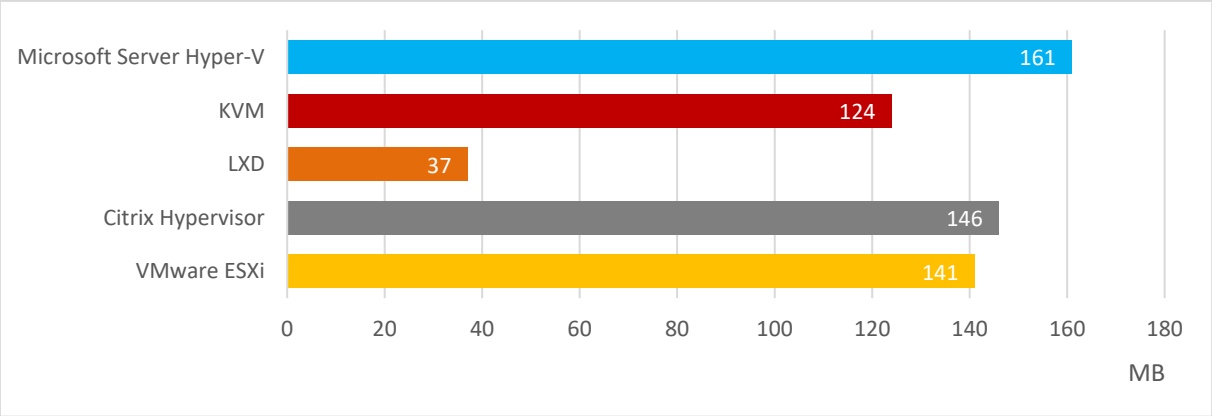
Pri druhom teste bola celková veľkosť dát určených k prenosu rovnaká, ako v prvom teste, teda 10GB. Zmenila sa veľkosť vyrovnávacej pamäte na 1kB, to znamená, že Sysbench musel vykonať daľeko viac operácií, ako v prvom prípade. Na Grafe 9 môžeme pozorovať dopad tejto zmeny pri jednotlivých platformách. Prekvapujúce výsledky v negatívnom zmysle môžeme pozorovať u platformy Citrix Hypervisor, ktorá mala v tomto teste značné problémy. Pokles prenosovej rýchlosti oproti fyzickej inštalácií o 82% bol naozaj markantný.



Graf 9: Sysbench RAM, 1kB vyrovnávacia pamäť.

8.2.3 Využitie a správa operačnej pamäte

Na záver som sa ešte zameral na využitie a správu operačnej pamäte. Najprv som sa zameral na využitie RAM pri hosťoch. Pri každej platforme som si najskôr vytvoril nový virtuálny stroj/kontajner s 2048MB RAM a následne pomocou nástroja htop odsledoval jej využitie. Ako môžeme vidieť na Grafe 10, potvrdila sa menšia náročnosť LXD kontajnerov, ktorou som sa zaoberal v teoretickej časti. Pri ostatných platformách bol rozdiel minimálny.

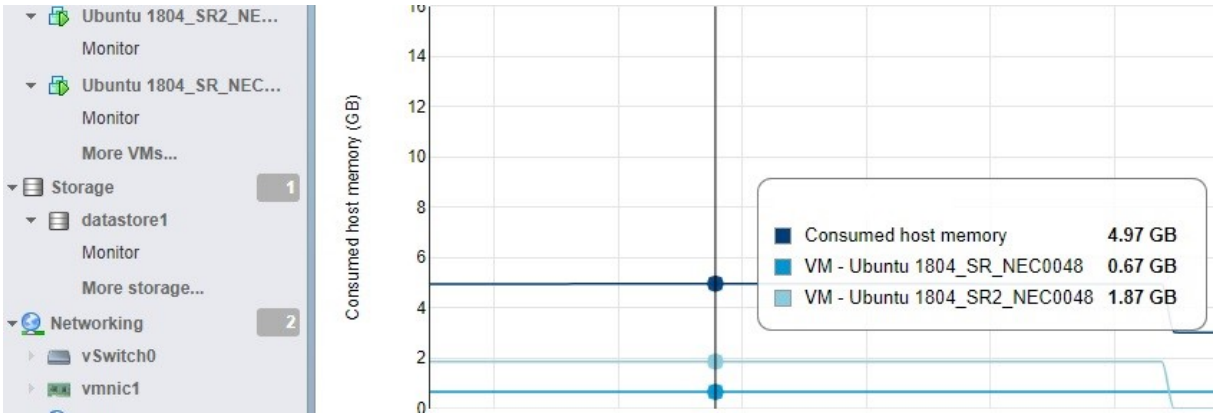


Graf 10: Využitie RAM pri novovytvorených hosťoch

Zaujímavejšie bolo sledovať, ako riešia správu RAM hostitelia. V podstate sú dve možnosti, ako môže hypervizor spravovať operačnú pamäť. Prvou možnosťou je, že hypervizor virtuálnemu stroju pri spustení alokuje automaticky plnú veľkosť pridelenej operačnej pamäte. Druhou možnosťou je alokácia len určitej veľkosti z pridelenej pamäte a následná dynamická správa, ktorá záleží hlavne od zaťaženia virtuálneho stroja a je v réžií hypervizora. Ďalšou možnosťou dynamickej správy je takzvaný „hot-plug", prípadne „hot-remove", ktorý je zasa v réžií administrátora a umožňuje za chodu virtuálneho stroja spravovať operačnú pamäť.

8.2.3.1 VMware ESXi

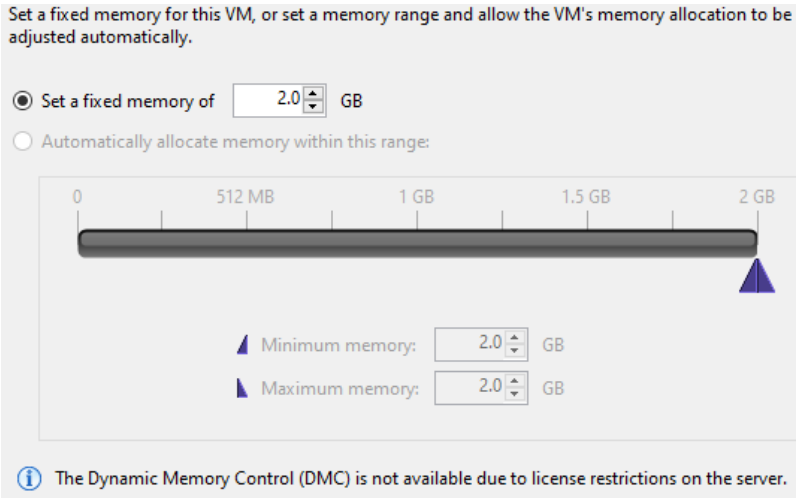
Dynamická správa operačnej pamäte je dostupná aj v samostatnom hypervizorovi VMware ESXi, čo je určite výhoda[75]. Správa operačnej pamäte je plne v réžií hypervizora. Podporovaný je aj spomenutý „hot-plug", ktorý ale nie je dostupný v tejto volne dostupnej verzii. Príklad takejto dynamickej správy operačnej pamäte môžeme pozorovať na Obrázku 50, kde som pridelil dvom virtuálnym strojom po 2048MB operačnej pamäte s tým, že virtuálny stroj Ubuntu 1804_SR_NEC0048 mal povolenú a nakonfigurovanú dynamickú správu a druhý, Ubuntu 1804_SR2_NEC0048 nie.



Obrázok 50: VMware ESXi - Dynamická správa operačnej pamäte

8.2.3.2 Citrix Hypervisor

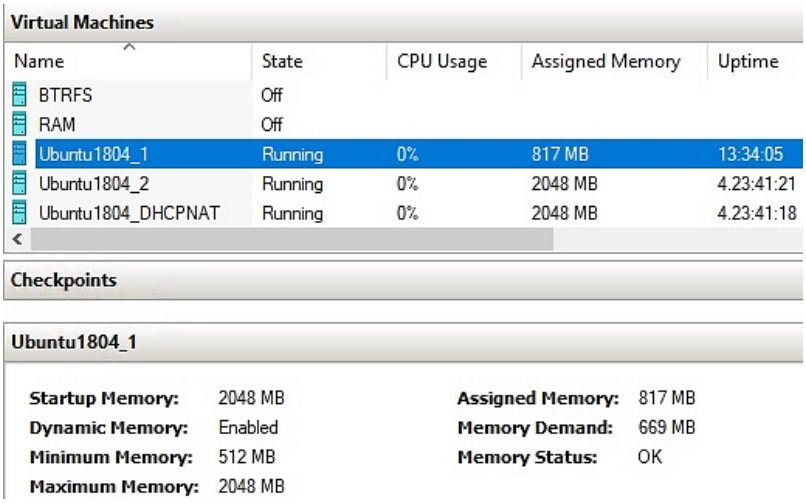
Verzia Citrix Hypervisor Express Edition, ktorá bola v tejto práci využitá a je dostupná zdarma žial neponúka možnosť dynamickej správy operačnej pamäte. Vo vyšších verziách je táto funkcia dostupná ako DMC (Dynamic Memory Control)[76]. DMC, inak aj „Memory balooning" funguje na podobnom princípe, ako pri VMware ESXi. Automaticky reguluje operačnú pamäť medzi špecifikovaným minimom a maximom. Pri mnou využitej verzií Express Edition alokoval hypervizor virtuálnemu stroju ihneď plnú pridelenú veľkosť operačnej pamäte. Na Obrázku 51 môžeme pozorovať, nastavenie DMC aj s oznámením o nedostupnosti kvôli licencií.



Obrázok 51: Citrix Hypervisor - Dynamická správa operačnej pamäte

8.2.3.3 *Microsoft Server Hyper-V*

Aj v prípade Hyper-V je dostupná dynamická správa operačnej pamäte. Prvý krát bola uvedená so serverovým operačným systémom Windows Server 2008 R2 SP1[77]. Podobne, ako predchádzajúce riešenia umožňuje flexibilne spravovať operačnú pamäť. Na Obrázku 52 môžeme pozorovať nastavenie dynamickej pamäte pri virtuálnom stroji Ubuntu1804_1 a momentálnu pridelenú RAM.

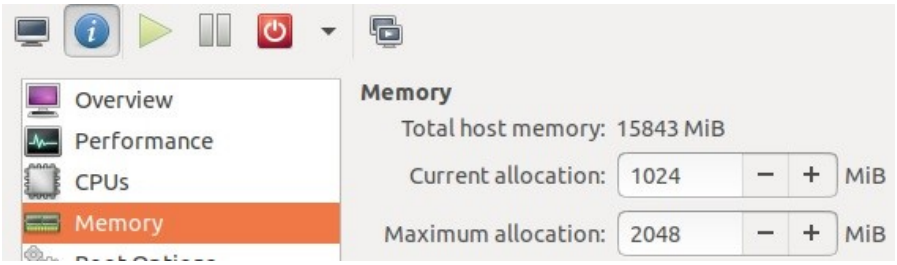


Obrázok 52: Microsoft Server Hyper-V - Dynamická správa operačnej pamäte

8.2.3.4 *KVM*

Aplikácia virt-manager ponúka nastavenie momentálnej a maximálnej alokácie z pridelenej operačnej pamäte pri KVM hosťoch. Avšak, nejedná sa o dynamickú správu v réžií hypervizora, jedná sa o už spomenutý „hot-plug" a „hot-remove" v réžií administrátora. To znamená, že administrátor môže flexibilne regulovať operačnú pamäť bez nutnosti vypnutia virtuálneho stroja. Ďalšou možnosťou je takzvaný „balloon driver"[78], ktorý je ale kontrolovaný pomocou virsh príkazov.

Na Obrázku 53 môžeme vidieť príklad konfigurácie v aplikácií virt-manager, maximálnu veľkosť som alokoval na 2048MiB RAM, ale ak bolo potrebné, mohol som priamo za behu virtuálneho stroja flexibilne regulovať operačnú pamäť cez „Current allocation".



Obrázok 53: KVM - Dynamická správa operačnej pamäte

8.2.3.5 ***LXD***

V prípade LXD som sa stretol len s dynamickou správou, kde pomocou už známeho lxc príkazu limits.memory pridelíme kontajneru maximálny limit operačnej pamäte. Zobraziť momentálnu veľkosť alokovanej operačnej pamäte môžeme prostredníctvom príkazu:

- `lxc info nazovkontajnera`

Na Obrázku 54 môžeme pozorovať výpis o stave operačnej pamäte, po zadaní príkazu lxc info.

```
Memory usage:
  Memory (current): 363.38MB
  Memory (peak): 2.33GB
```

Obrázok 54: LXD - Dynamická správa operačnej pamäte

8.2.4 Zhrnutie

Keď sa pozrieme na základný test RAMspeed SMP a výsledky z Grafu 6, tak môžeme povedať že v prípade nezaťaženého procesoru podávali jednotlivé virtualizačné platformy stabilné a relatívne vyrovnané výsledky. V prípade kontajnerovej platformy LXD sme v niekoľkých prípadoch hovorili dokonca o mierne lepších výsledkoch, ako pri samotnej fyzickej inštalácií Ubuntu Server 18.04.3 LTS. Na druhú stranu, prekvapujúce sú výsledky hlavne posledného testu Sysbench z Grafu 9, kde mal Citrix Hypervisor v prípade alokácie vyrovnávacej pamäti vo veľkosti 1kB oproti ostatným platformám markantné problémy.

Pri správe operačnej pamäte bolo zaujímavé sledovať, ako je riešená hlavne dynamická správa z pohľadu hypervizora.

8.3 Test výkonu súborového systému

Pre test výkonu súborového systému boli zvolené dva porovnávacie nástroje, IOzone[73] a Unpacking The Linux Kernel[74]. Obidva nástroje sú podobne, ako niektoré predchádzajúce súčasťou platformy Phoronix Test Suite.

Pred samotným testovaním si bolo potrebné uvedomiť jeden podstatný fakt, ktorý odlišuje kontajnerovú platformu LXD od ostatných použitých virtualizačných platforiem, hovoríme o použitom type súborového systému. Nakoľko pri LXD neinštalujeme kompletný operačný systém, tak typ súborového systému, ktorý budú využívať jednotlivé kontajnery závisí na konfigurácií úložiska príkazom lxd init. Podrobnejšie som sa tejto problematike venoval v kapitole 5.6.1, respektíve 5.6.2. Pri ostatných virtualizačných platformách, VMware ESXi, Citrix Hypervisor, Microsoft Hyper-V a KVM inštalujeme kompletný systém, tak ako na klasický fyzický počítač. Ďalším faktorom je súborový systém nie hosťa, ale hostiteľa.

Ja som pojal tento test mierne odlišne. Pri plnohodnotných virtuálnych strojoch som sa rozhodol zaradiť dva súborové systémy a sledovať vzájomné rozdiely nie len medzi platformami, ale aj vplyv výkonu na typ súborového systému. Pri LXD kontajneroch som zvolil len jeden súborový systém a to BTRFS. Pre plnohodnotné virtuálne stroje som zvolil EXT4 a BTRFS, oba súborové systémy sú plnohodnotne podporované použitým operačným systémom Ubuntu Server 18.04.3 LTS.

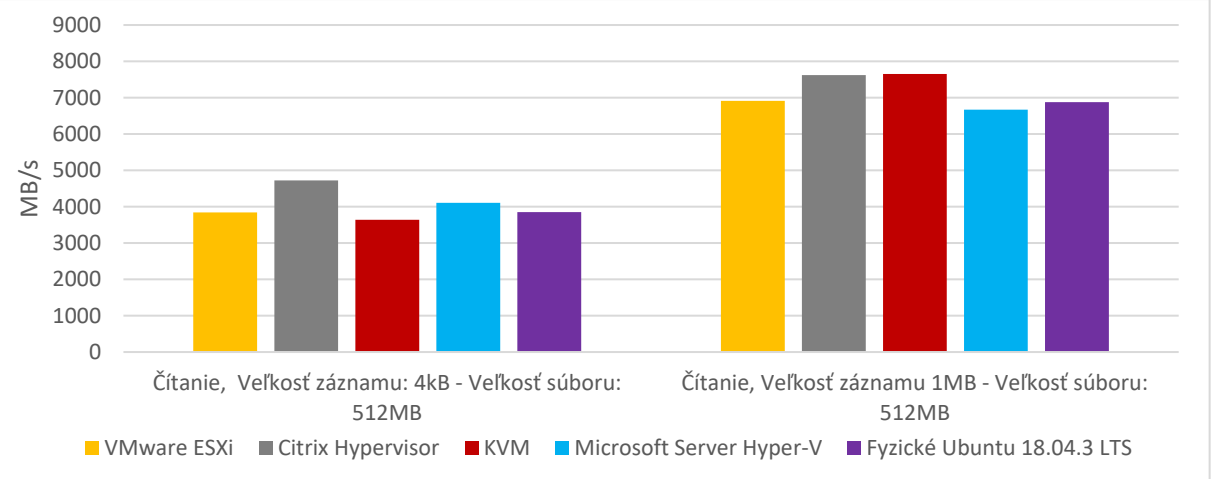
Pre testovanie som zvolil dve konfigurácie nástroja IOzone. Prvá konfigurácia zahŕňala nastavenie veľkosti záznamu na 4kB a veľkosť súboru 512MB. Druhá konfigurácia zahŕňala nastavenie veľkosti záznamu na 1MB, veľkosť súboru zostala nezmenená, teda 512MB. Ako doplnkový test bol zvolený Unpacking The Linux Kernel, tento test zaznamenáva dobu potrebnú na extrahovanie archívu .tar.xz obsahujúceho balíček Linux jadra vo verzií 4.15.

Počas obidvoch testov boli ostatné virtuálne stroje vypnuté. Každý test prebehol trikrát, výsledná a použitá hodnota bol aritmetický priemer nameraných výsledkov.

8.3.1 IOzone - EXT4

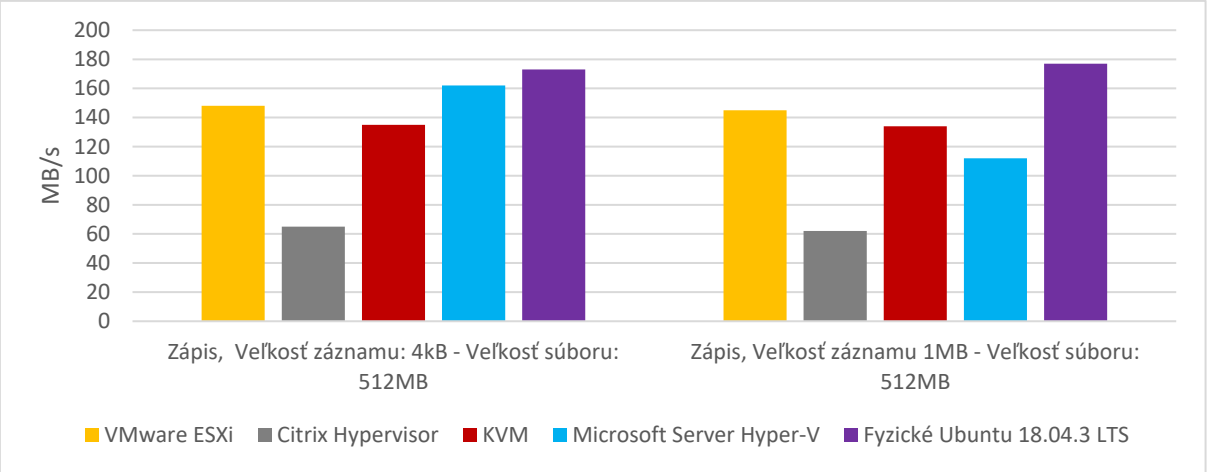
Pri prvom teste som sa zameral na výkon súborového systému EXT4 a vzájomné porovnanie medzi virtualizačnými platformami a fyzickou inštaláciou. Tento test nezahŕňal platformu LXD.

Zaujímavé výsledky môžeme pozorovať na Grafe 11, kde pri niektorých platformách môžeme pozorovať vyšší výkon v čítaní, ako pri fyzickej inštalácií. Pri veľkosti záznamu 1MB podali platformy Citrix Hypervisor a KVM oproti fyzickej inštalácií lepší výsledok približne o 10%.



Graf 11: IOzone Čítanie - EXT4

Pri teste zápisu môžeme pozorovať vyššie výkonnostné rozdiely. Na Grafe 12 môžeme vidieť prepad platformy Citrix Hypervisor oproti fyzickej inštalácií približne o 65%. Taktiež si môžeme všimnuť, že veľkosť záznamu nijak zásadne neovplyvnila výsledky, tie boli pri všetkých platformách takmer identické, výnimku tvoril len Microsoft Hyper-V.

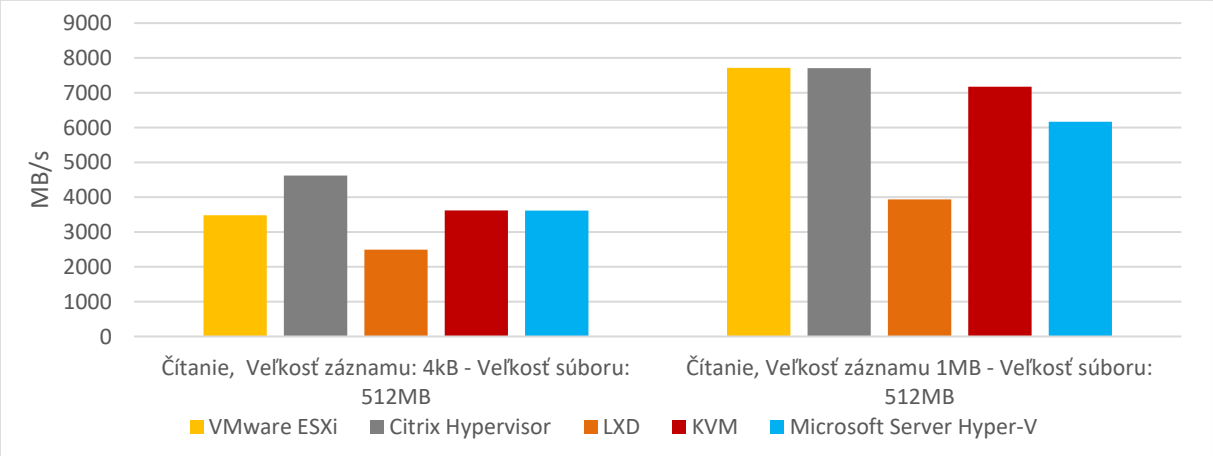


Graf 12: IOzone Zápis - EXT4

8.3.2 IOzone - BTRFS

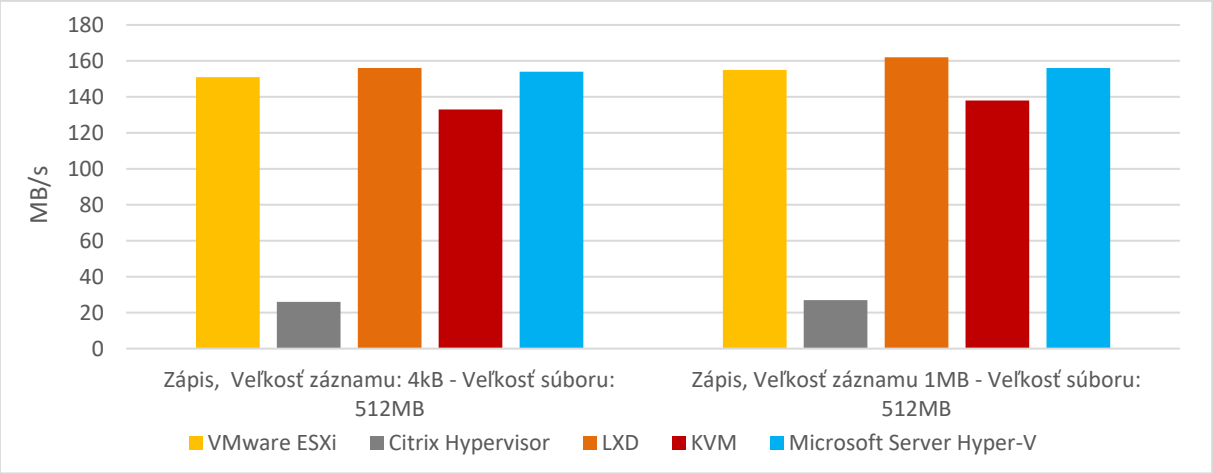
K tomuto testu mi poslúžili rovnako nakonfigurované virtuálne stroje, ako pri predchádzajúcom teste, ale s tým, že namiesto súborového systému EXT4, boli jednotlivé virtuálne stroje vytvorené so súborovým systémom BTRFS. Pri tomto teste bola namiesto fyzickej inštalácie, ktorá využívala EXT4 zaradená platforma LXD.

Na Grafe 13 môžeme vidieť výsledky, ktoré sa výrazne nelíšia od tých z identického testu na Grafe 10. V porovnaní s EXT4 môžeme pozorovať pri veľkosti záznamu 1MB pokles výkonu u platformy KVM, naopak vyšší nárast výkonu môžeme pozorovať u VMware ESXi. Prekvapujúci prepad približne o 50% oproti Citrix Hypervisor zaznamenala platforma LXD, ktorá doteraz nemala väčšie problémy.



Graf 13: IOzone Čítanie - BTRFS

Pri teste zápisu na Grafe 14 môžeme pozorovať, že platforma Citrix Hypervisor zaznamenala ešte horšie výsledky zápisu, ako pri súborovom systéme EXT4, hovoríme o poklese približne o 68% v porovnaní s Grafom 11. Za pozornosť určite stojí platforma LXD, ktorá pri tomto teste podávala najlepšie výsledky, jedná sa o zaujímavý kontrast s Grafom 12, kde sme hovorili o presnom opaku.

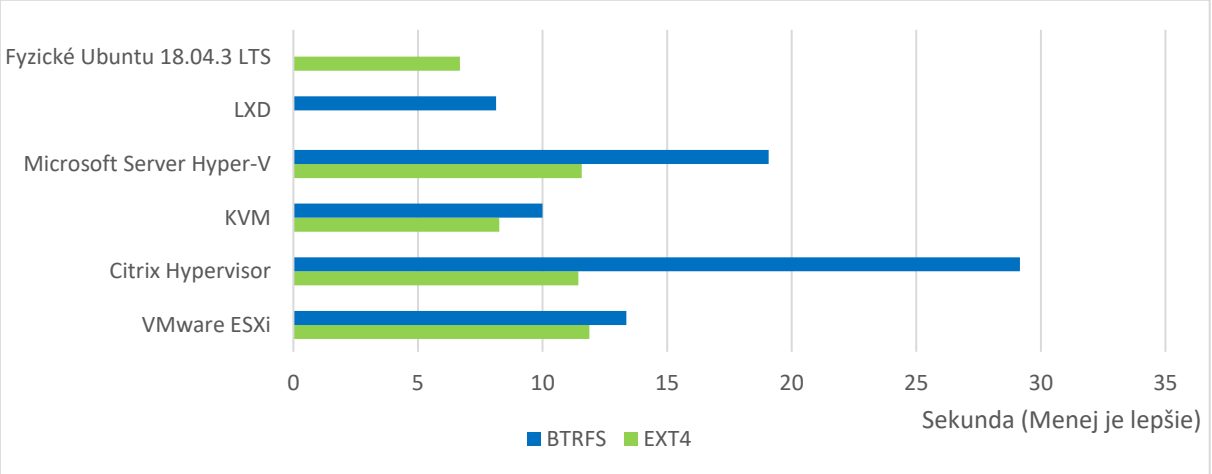


Graf 14: IOzone Zápis - BTRFS

8.3.3 Unpacking The Linux Kernel

Pri tomto doplnkovom teste na Grafe 15 môžeme pozorovať, že tri platformy VMware ESXi, Citrix Hypervisor a Microsoft Hyper-V zazamenali v prípade použitia súborového systému EXT4 takmer identické výsledky. Najlepšie si v teste EXT4 viedla platforma KVM, ktorá zaznamenala oproti fyzickej inštalácií horší výsledok o približne 20%.

Naopak, v prípade testu BTRFS nastali podstatné rozdiely. Citrix Hypervisor mal opäť najvačšie problémy so stabilitou výkonnosti, oproti platfrome LXD zaostal o približne 72%. Najlepšie výsledky spomedzi platforiem zastupujúcich klasické virtuálne stroje môžeme opäť pozorovať u KVM, ktorá zaostala za LXD o približne 19%.



Graf 15: Unpacking The Linux Kernel

8.3.4 Zhrnutie

V prípade klasických plnohodnotných virtuálnych strojov sme mohli pozorovať, že voľba súborového systému v podobe EXT4 alebo BTRFS nemala nejaký dramatický vplyv na výkonnosť. Jedinu výnimku tvoril Citrix Hypervisor, ktorý mal problémy hlavne v testoch zápisu, či už sa jednalo o záznam veľkosti 4kB alebo 1MB. Osobne si však myslím, že väčšiu úlohu hral súborový systém hostiteľa, prípadne fragmentácia použitých HDD, ktorá v tomto prípade mohla mať vplyv na výsledky.

Pri virtuálnych prostrediach LXD som k testom využil úložisko so súborovým systémom BTRFS. Oproti plnohodnotným virtuálnym strojom boli výsledky v čítaní horšie približne o 50%, naopak pri zápise sme pozorovali najlepšie výsledky spomedzi všetkých platforiem.

8.4 Test virtuálnej siete

Pre testy virtuálnych sietí boli zvolené nástroje iPerf3[79] a PING[80]. IPerf3 je zameraný na meranie maximálnej dosiahnutelnej prenosovej rýchlosti v IP sieťach. Umožňuje nastavenie rôznych parametrov týkajúcich sa načasovania a protokolov TCP, UDP alebo SCTP. Pri testovaní som sa zameral jak na protokol TCP, tak aj na UDP. Najskôr som sa zameral na maximálnu možnú prenosovú rýchlosť dát medzi virtuálnymi strojmi pre obidva protokoly, teda TCP a UDP. Ďalší test bol zameraný na vplyv veľkosti TCP okna na prenosovú rýchlosť. Známy nástroj PING slúžil k zisteniu odozvy na správy ECHO request a ECHO reply protokolu ICMP.

K testom som využil vždy dva virtuálne stroje v rámci hostiteľa, kde jeden slúžil ako server a druhý ako klient. Každý test bol spustený päťkrát, výsledná a použitá hodnota bol aritmetický priemer nameraných výsledkov. Pre prehladnosť som uvádzal pri testoch iPerf3 aj jednotlivé príkazy.

V tejto kapitole som sa ďalej zameral na vplyv prevádzky vo virtuálnej sieti na virtualizačnú platformu. K testu mi vždy slúžili dva virtuálne stroje v rámci hostiteľa s pridelenými dvomi CPU a 2048MB RAM. Prvý virtuálny stroj slúžil ako webový server postavený na NGINX[83], na ktorom bola spustená moderná monitorovacia platforma NETDATA[81]. Druhý virtuálny stroj slúžil ako generátor HTTP záťaže pomocou nástroja wrk[82].

8.4.1 iPerf3 - Test TCP

Prvý test v rámci TCP bol zameraný na maximálnu možnú prenosovú rýchlosť dát medzi virtuálnymi strojmi.

Príkaz na spustenie serveru pre príjem TCP spojení na prednastavenom porte 5201:

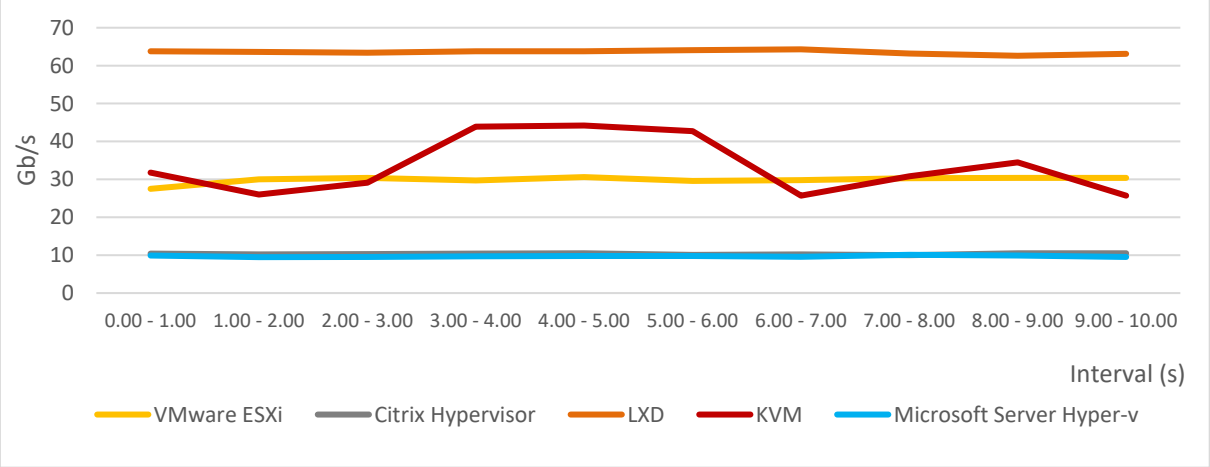
- `iperf3 -s`

Príkaz na strane klienta pre generovanie TCP spojení:

- `iper3 - c ip_adresa_serveru`

Dobu testu a interval generovania TCP spojení som ponechal v prednastavenej hodnote, teda dĺžka testu 10 sekúnd a interval 1 sekunda. Ako môžeme vidieť na Grafe 16, suverénne najvyššiu prenosovú rýchlosť v podobe stabilných 63Gb/s dosahovala platforma LXD. Platformy VMware ESXi a KVM dosiahli prenosové rýchlosti približne 30Gb/s, respektíve 33Gb/s.

Pri platforme KVM však môžeme pozorovať horšiu stabilitu v prenosovej rýchlosti medzi jednotlivými intervalmi počas testu. Takmer identické a stabilné prenosové rýchlosti približne 10Gb/s dosiahli platformy Microsoft Hyper-V a Citrix Hypervisor.

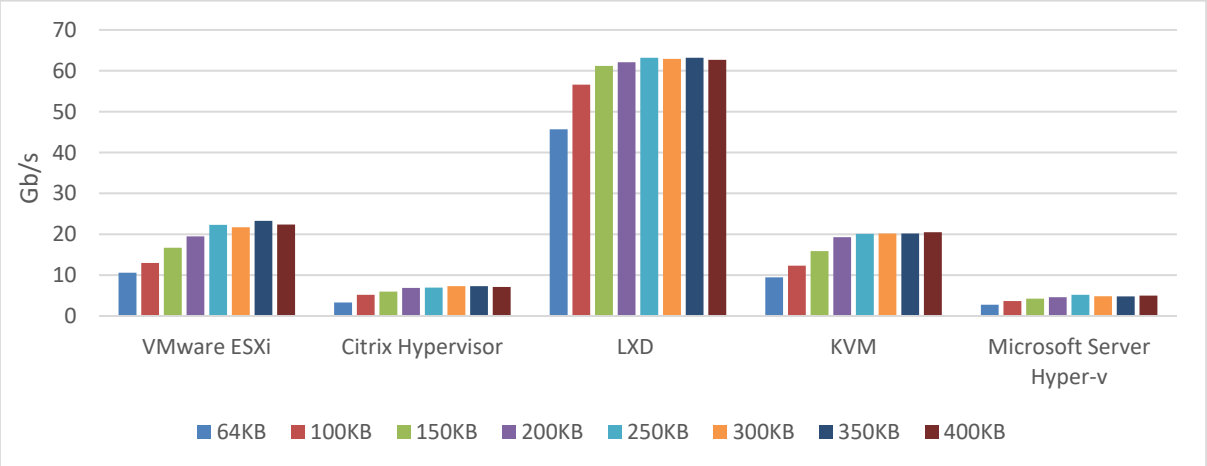


Graf 16: TCP - Test maximálnej možnej prenosovej rýchlosti medzi virtuálnymi strojmi

Druhý test bol zameraný na vplyv veľkosti TCP okna na prenosovú rýchlosť. Použil som identické príkazy, ako pri predchádzajúcom teste s tým, že príkaz na strane klienta bol doplnený o parameter špecifikujúci veľkosť TCP okna:

- `iperf3 -c ip_adresa_serveru -w sirka_tcp_okna_KB`

Na Grafe 17 môžeme pozorovať, že maxima v prenosovej rýchlosti začali jednotlivé platformy dosahovať približne pri okne vo veľkosti 200KB, respektíve 250KB.



Graf 17: Vplyv veľkosti TCP okna na prenosovú rýchlosť

8.4.2 iPerf3 - Test UDP

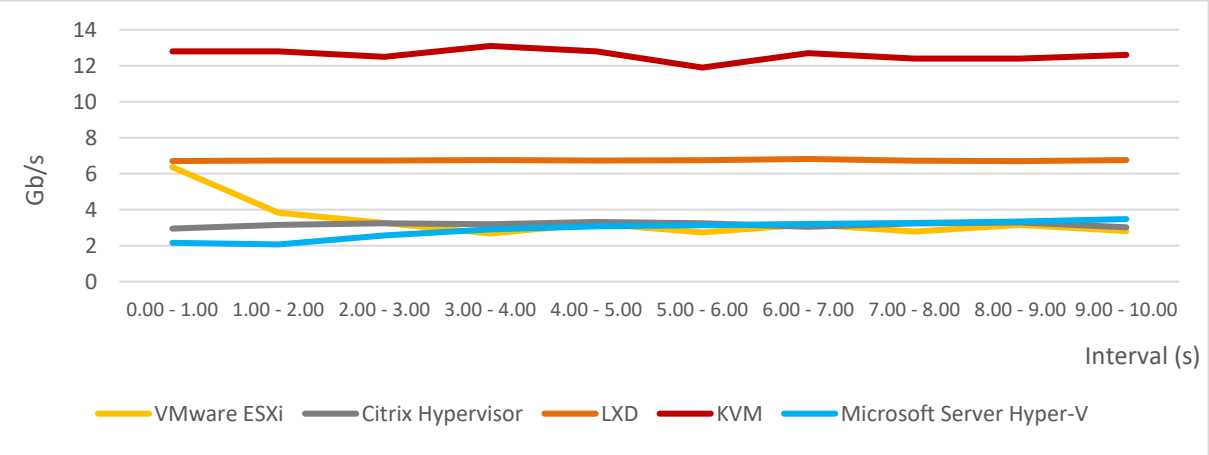
Podobne, ako pri TCP bol test zameraný na maximálnu možnú prenosovú rýchlosť dát medzi virtuálnymi strojmi. Príkaz na spustenie serveru pre príjem UDP spojení na prednastavenom porte 5201 bol identický s TCP.

Príkaz na strane klienta pre generovanie UDP spojení:

- `iperf3 -c ip_adresa_serveru -u -b 0`

Parameter -u značí použitie UDP protokolu, -b 0 značí nastavenie neobmedzenej šírky pásma. V prípade, že by sme test UDP spustili bez parametru -b 0, tak by bola rýchlosť automaticky limitovná len na 1,05Mb/s.

Ako môžeme pozorovať na Grafe 18, najvyššiu prenosovú rýchlosť v priemere 12,5Gb/s dosiahla platforma KVM, ale to za cenu obrovskej strátovosti datagramov v Tabuľke 6. Osobne som si po prvom teste myslel, že sa jedná o chybný test, ale po vykonaní niekoľko testov boli výsledky stále takmer identické. Za najlepšie výsledky môžeme považovať tie od platformy LXD, ktorá zaznamenala prenosovú rýchlost v priemere 6,7Gb/s a žiadnu strátovosť datagramov. Jitter, ktorý signalizuje preťaženie siete bol pri všetkých platformách nízky.



Graf 18: UDP - Test maximálnej možnej prenosovej rýchlosti medzi virtuálnymi strojmi

Tabuľka 5: Doplnok ku Grafu 18 - Jitter a strátovosť datagramov

	Jitter (ms)	Stratený / Celkový počet datagramov
VMware ESXi	0,010	4751 / 519210 (0,92%)
Citrix Hypervisor	0,014	15187 / 4824449 (3,1%)
LXD	0,002	0 / 1028340 (0%)
KVM	0,134	1680202 / 1921833 (87%)
Microsoft Server Hyper-V	0,004	32139 / 455672 (7,2%)

8.4.3 **Test odozvy pomocou nátroja PING**

Tento test bol zameraný na takzvaný „roundtrip time", ktorý znamená, ako dlho trvalo paketu dosiahnuť cieľ a vrátiť sa späť k odosielateľovi. Pri každom z testov bolo vygenrovaných desať ECHO request paketov. Sledoval som minimálnu, maximálnu, priemernú odozvu, smerodajnú odchýlku a straty. Výsledky môžeme sledovať v Tabuľke 6.

Tabuľka 6: PING - Test odozvy

	Minimum (ms)	Maximum (ms)	Priemer (ms)	Smerodajná odchýlka (ms)	Straty (%)
VMware ESXi	0,080	0,106	0,091	0,008	0
Citrix Hypervisor	0,194	0,445	0,350	0,066	0
LXD	0,036	0,038	0,037	0,002	0
KVM	0,303	0,627	0,495	0,091	0
Microsoft Hyper-V	0,130	0,217	0,161	0,027	0

8.4.4 **Vplyv prevádzky vo virtuálnej sieti na serverovú platformu**

Ako som už spomenul v úvode kapitoly, k testovaniu som použil virtuálny stroj s webovým serverom NGINX a predvolenou konfiguráciou. Monitorovanie tohoto virtuálneho stroja zabezpečila moderná platforma NETDATA.

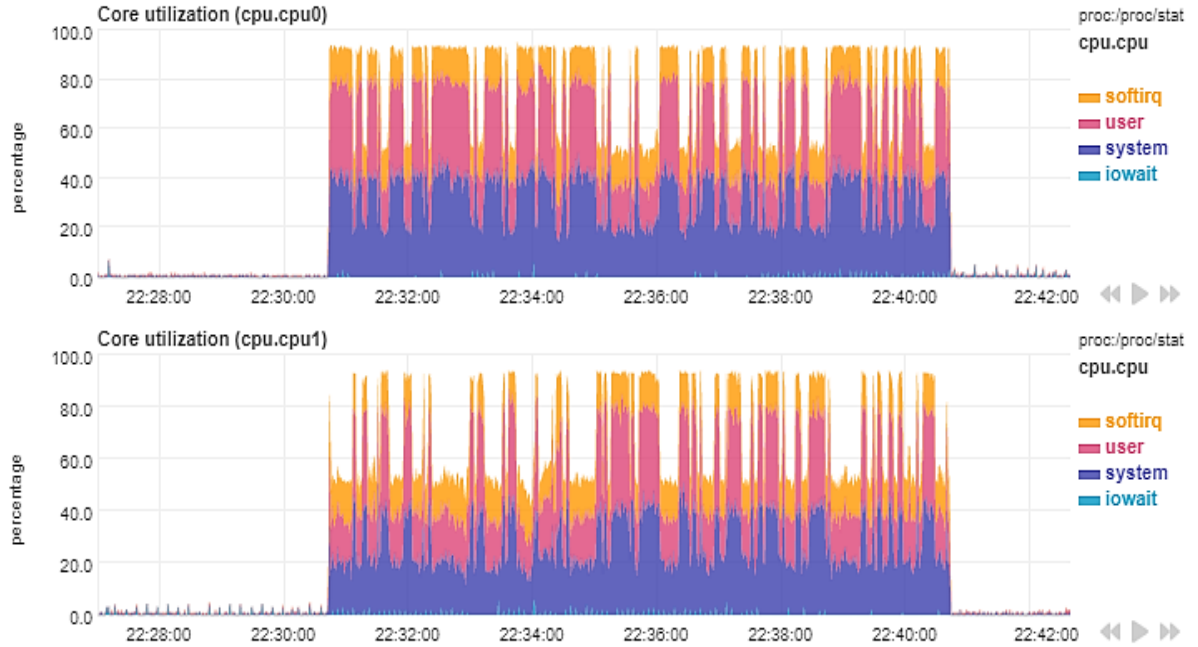
Test spočíval v generovaní a udržovaní 50 HTTP spojení po dobu 10 minút pri použití dvoch vlákien pomocou nástroja wrk, konkrétny príkaz:

- `wrk -t2 -c50 -d10m http://ip_adresa/index.html`

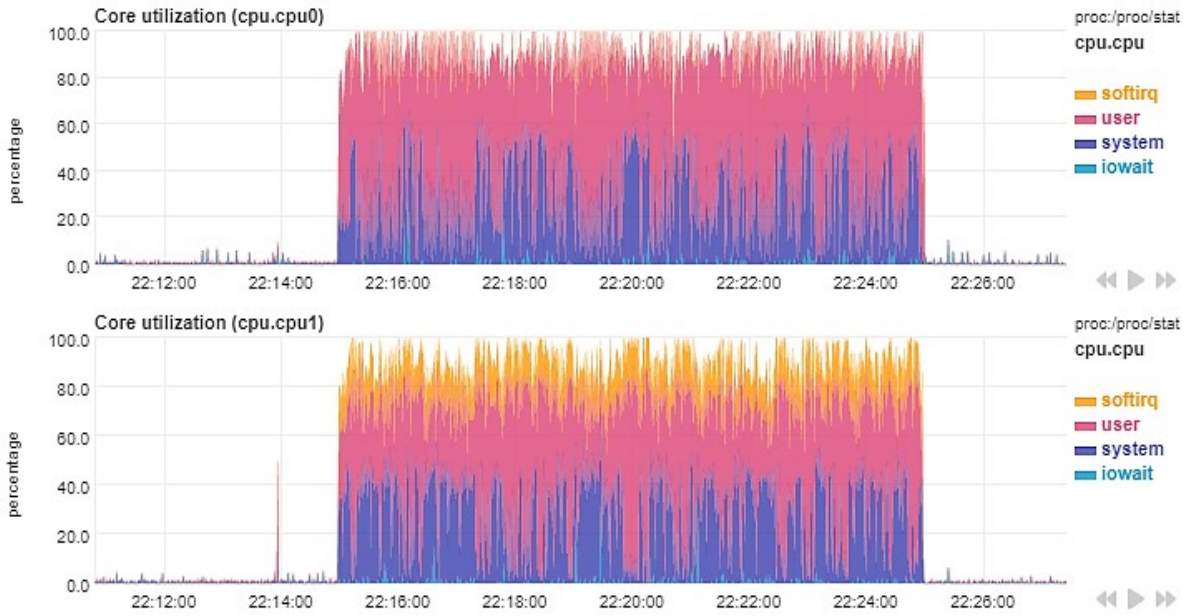
Na Obrázkoch 55, 56, 57, 58 a 59 môžeme pozorovať priebeh vyťaženia CPU virtuálnych strojov s webovým serverom NGINX. Priemerné hodnoty vyťaženia sa pohybovali v závislosti na platforme medzi 75 - 90%.

V prílohe B sú priložené kompletné súbory, ktoré obsahujú nespočet ďalších štatistík z týchto testov. Príloha mimo týchto súborov obsahuje aj návod, ako si tieto štatistiky prehľadne zobraziť priamo vo webovom prehliadači.

Obrázok 55: Využitie CPU - Citrix Hypervisor

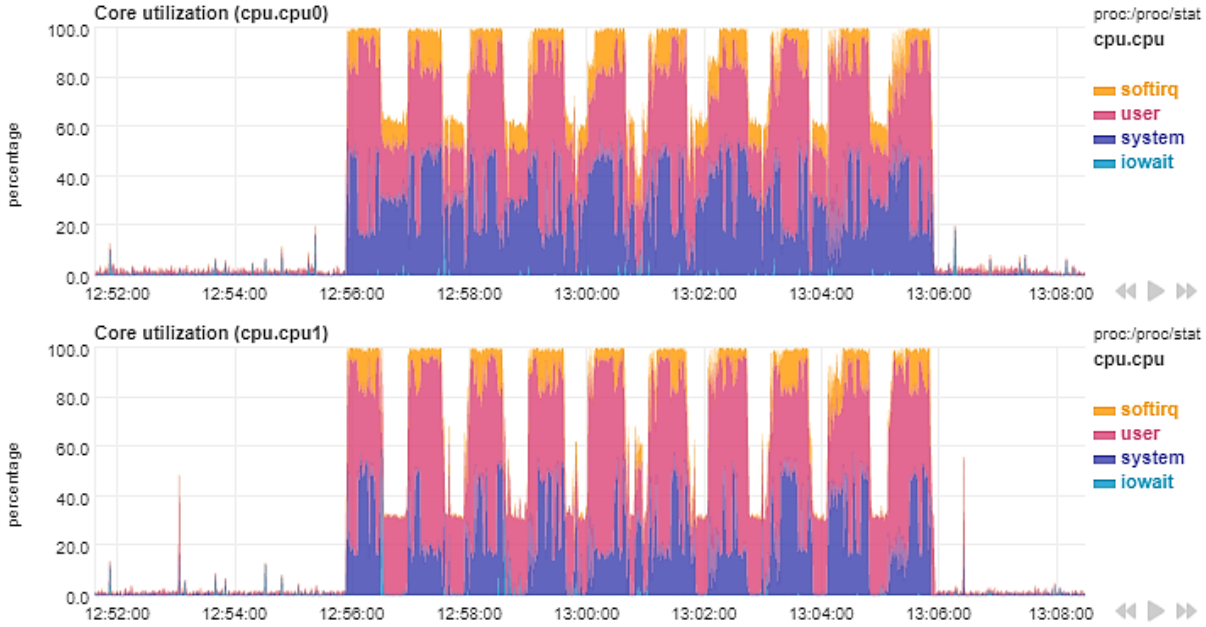


Obrázok 56: Využitie CPU - VMware ESXi



Obrázok 57: Využitie CPU - Microsoft Hyper-V

Obrázok 58: Využitie CPU - KVM



Obrázok 59: Využitie CPU - LXD

8.4.5 **Zhrnutie**

Ako sme mohli sledovať hlavne pri testoch nástroja iPerf3, tak takmer každá platforma pri využití svojej implementácie virtuálneho prepínača a virtuálneho sieťového adaptéra podávala rozdielne výkonové výsledky. Napríklad v teste protokolu TCP podával VMware ESXi podstatne lepšie výsledky, ako takmer identicky nakonfigurované platformy Citrix Hypervisor a Microsoft Hyper-V. Osobne si myslím, že za rozdiel môže pokročilý virtuálny sieťový adaptér VMXNET3, ktorý VMware ESXi hostia štandardne využívajú. Taktiež som prekvapený z výsledkov LXD, pri tejto platforme som očakával výsledky niekde na úrovni KVM, ktoré síce zaznamenalo najvyššiu maximálnu prenosovú rýchlosť pri teste UDP, ale to za cenu veľkej strátovosti datagramov, kde naopak LXD zaznamenalo polovičnú prenosovú rýchlosť, ale žiadnu strátovosť.

9 Metodika testovania a porovnanie virtualizačných platforiem pre OS Linux a Windows

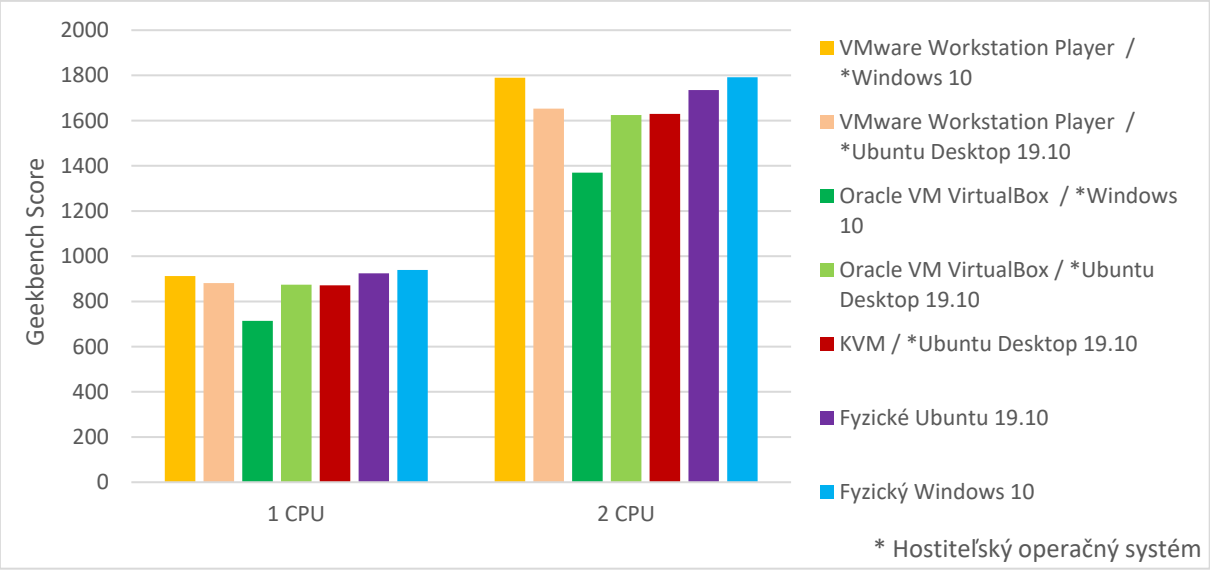
Podobne, ako pri serverových riešeniach som sa aj tu zameral hlavne na porovnanie výkonu jednotlivých platforiem. Ako som už spomenul v kapitole 7, základ všetkých virtuálnych strojov bol opäť použitý operačný systém Ubuntu Server 18.04.3 LTS. Niektoré testy zahŕňajú fyzickú inštaláciu Windows 10 alebo Ubuntu 19.10. Všetky testy boli vykonané pri čo najmenšom možnom zaťažení celého hostiteľského systému, takže programy, ktoré by mohli negatívne ovplyvniť výsledky boli počas jednotlivých testov vypnuté.

9.1 Test výkonu procesoru

Základom môjho prenosného počítača je štvorjadrový, osemvláknový procesor Intel Core i5-8250U s podporou technológie „Hyper-threading". Pre test výkonu procesoru boli využité porovnávacie nástroje Geekbench5 a Sysbench. Virtuálne stroje mali pridelené jedno alebo dve CPU a 2048MB RAM. Každý test prehebol trikrát, výsledná a použitá hodnota bol aritmetický priemer nameraných výsledkov.

9.1.1 Geekbench 5

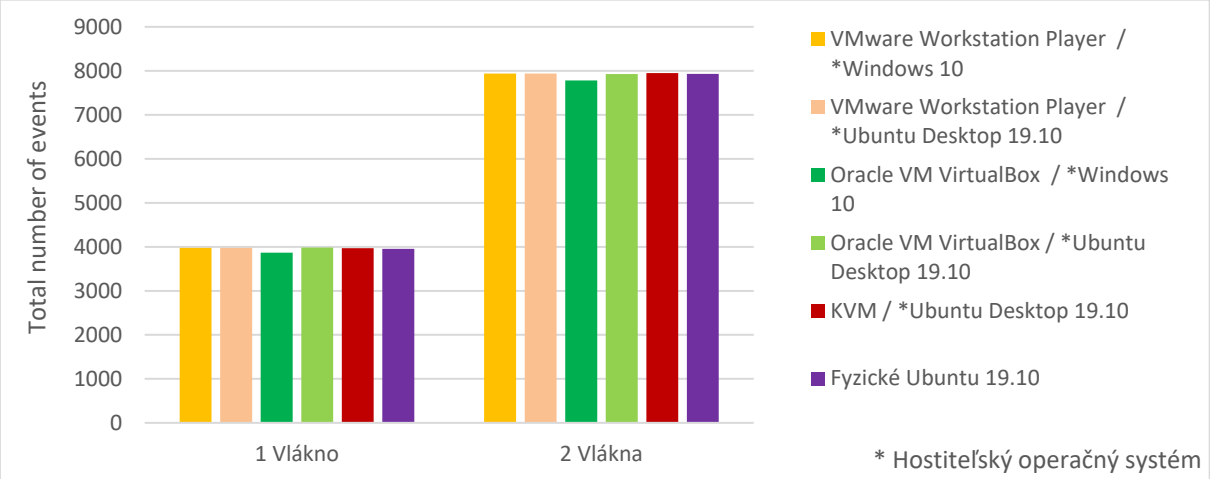
Graf 19 zobrazuje jednotlivé výsledky, kde najlepšie podávala platfroma VMware Workstation Player a to pri obidvoch hostiteľských OS. Prekvapením je Oracle VM VirtualBox, kde táto platforma zaostala oproti výsledkom VMware Workstation Player o približne 23% a to pri použití hostiteľského OS Windows 10, pri použití hositeľského OS Ubuntu 19.10 boli výsledky týchto platforiem takmer idenitické.



Graf 19: Geekbench5 CPU - Platformy pre OS Linux a Windows

9.1.2 Sysbench CPU

Pri teste CPU prostredníctvom nástroja Sysbench môžeme na Grafe 20 pozorovať vyrovnané výsledky jednotlivých platforiem. Zanedbateľný prepad v podobe 2% oproti fyzickej inštalácií Ubuntu 19.10 zaznamenal opäť Oracle VM VirtualBox s hostiteľským Windows 10, tento prepad ale nebol taký znateľný, ako pri predchádzajúcom teste na Grafe 19.



Graf 20: Sysbench CPU - Platformy pre OS Linux a Windows

9.1.3 Zhrnutie

Pri teste CPU môžeme pozorovať, že voľba hostiteľského operačného systému mala minimálny dopad na výsledky testov. Osobne som čakal pri hostiteľskom operačnom systéme Ubuntu 19.10 mierne lepšie výsledky KVM v porovnaní s ostatnými platformami a to z dôvodu jeho integrácie priamo do Linux jadra. Na druhú stranu, pri týchto testoch CPU sa ukázala sila hardvérovej virtualizácie a technológie Intel VT-x, ktorá umožnila platformám dosahovať takmer natívne výsledky.

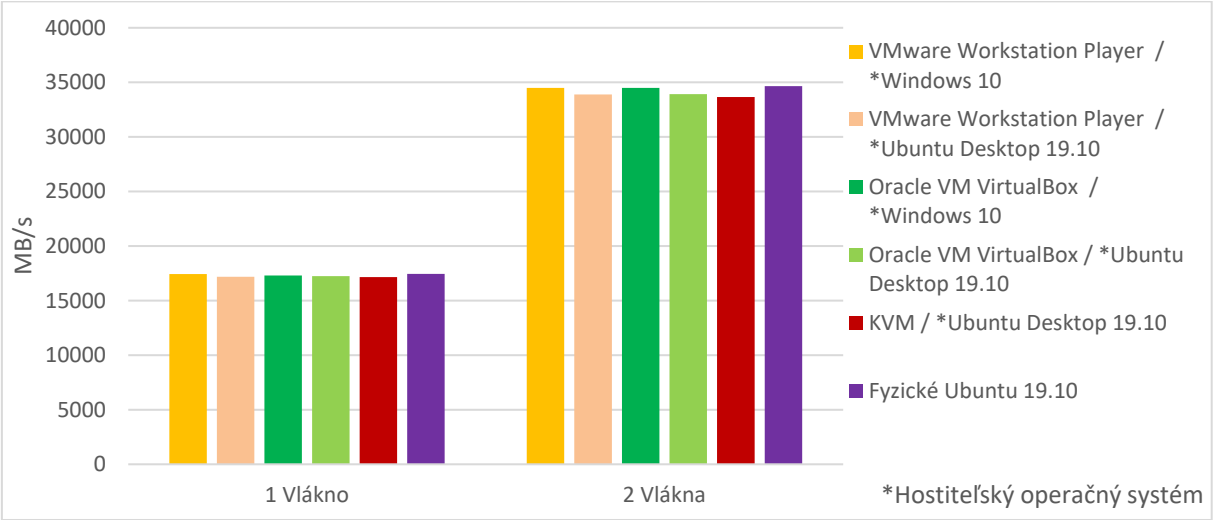
9.2 Test výkonu operačnej pamäte

Testy prebiehali identicky, ako v prípade serverovej virtualizácie, ale s tým, že bol použitý len porovnávací nástroj Sysbench. Virtuálne stroje mali počas tohoto testu pridelené jedno alebo dve CPU a 2048MB RAM, ostatné virtuálne stroje boli vždy vypnuté. Jediný parameter, ktorý som menil bola alokácia vyrovnávacej pamäte, konkrétne 1kB a 1MB.

Každý z testov bol spustený trikrát, výsledná a použitá hodnota bol artimetický priemer nameraných výsledkov.

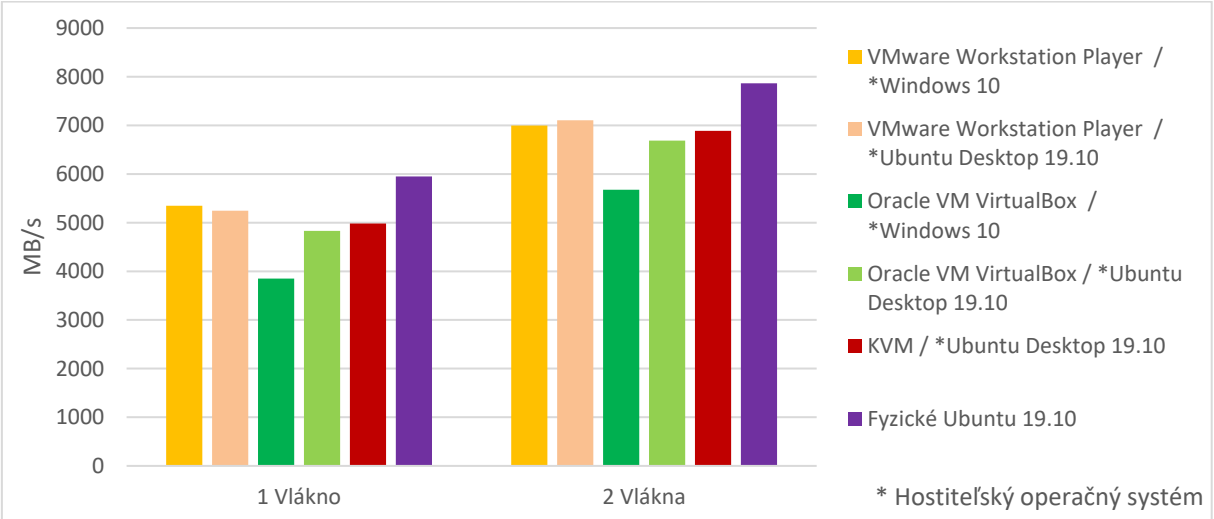
9.2.1 Sysbench RAM

Ako môžeme pozorovať na Grafe 21, pri vyrovnávacej pamäti o veľkosti 1MB boli výsledky jednotlivých platforiem veľmi podobné a v porovnaní s fyzickou inštaláciou Ubuntu Desktop 19.10 skoro aj natívne.



Graf 21: : Sysbench RAM, 1MB vyrovnávacia pamäť - Platformy pre OS Linux a Windows

Rozdiely môžeme pozorovať na Grafe 22 a pri vyrovnávacej pamäti o veľkosti 1kB, kde je znateľný pokles výkonu všetkých platforiem oproti fyzickej inštalácií. Najhoršie výsledky zaznamenala platforma Oracle VM VirtualBox pri použitom hostiteľskom OS Windows 10, hovoríme o približne 33% poklese oproti fyzickej inštalácií.



Graf 22: Sysbench RAM, 1kB vyrovnávacia pamäť - Platformy pre OS Linux a Windows

9.2.2 **Zhrnutie**

Prvý test, kde bola alokácia vyrovnávacej pamäti 1MB boli výsledky jednotlivých platforiem takmer identické. Pri druhom teste s alokáciou vyrovnávacej pamäti o veľkosti 1kB sme mohli pozorovať pokles výkonu všetkých platforiem oproti fyzickej inštalácií. Najvačší pokles výkonu sme mohli už po druhý krát pozorovať u platformy Oracle VM VirtualBox s hostiteľským Windows 10.

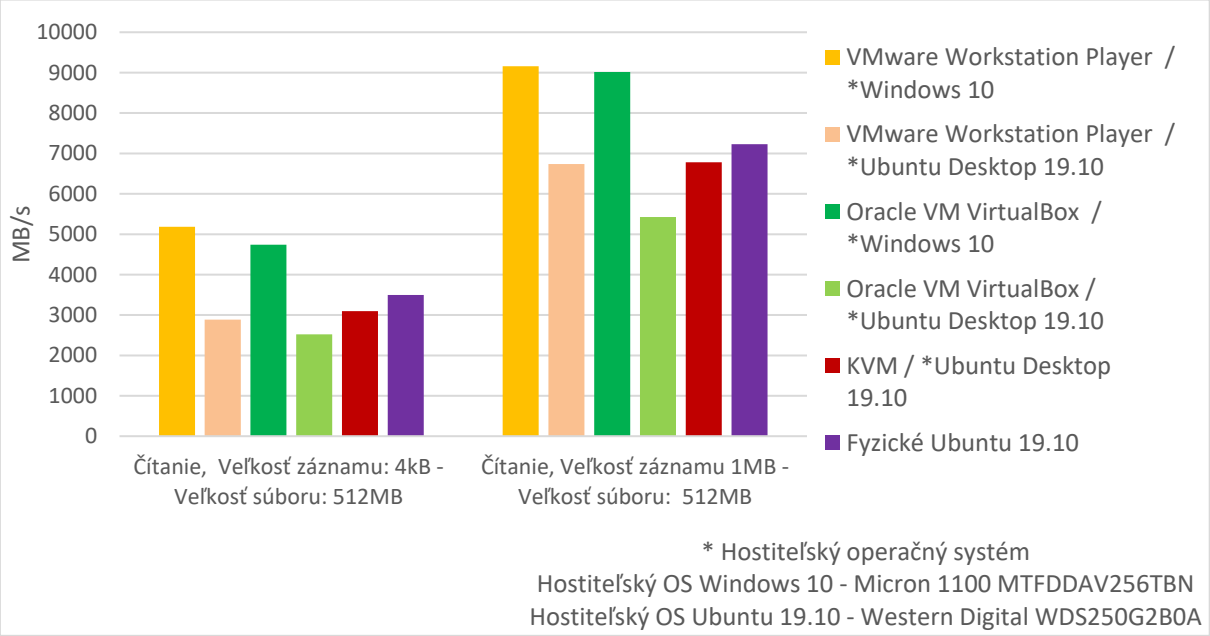
9.3 Test súborového systému

Pre test súborového systému bol zvolený porovnávací nástroj IOzone a jeho identická konfigurácia, ako v prípade serverových platforiem. Tentokrát boli jednotlivé virtuálne stroje vytvorené len so súborovým systémom EXT4. Ako som už spomenil v kapitole 7, na svojom prenosom počítači využívam izolovaný „dualboot", takže tým pádom boli v tomto teste využité dva rozdielne SSD disky. Kvôli tomuto faktu nie je možné porovnať výsledky jednotlivých platforiem pri použití rozdielneho hostiteľského OS. Pre prehľadnosť som v grafoch uviedol aj hostiteľský operačný systém s konkrétnym typom SSD disku, ktorý bol pre testy využitý.

Každý z testov bol spustený trikrát, výsledná a použitá hodnota bol artimetický priemer nameraných výsledkov.

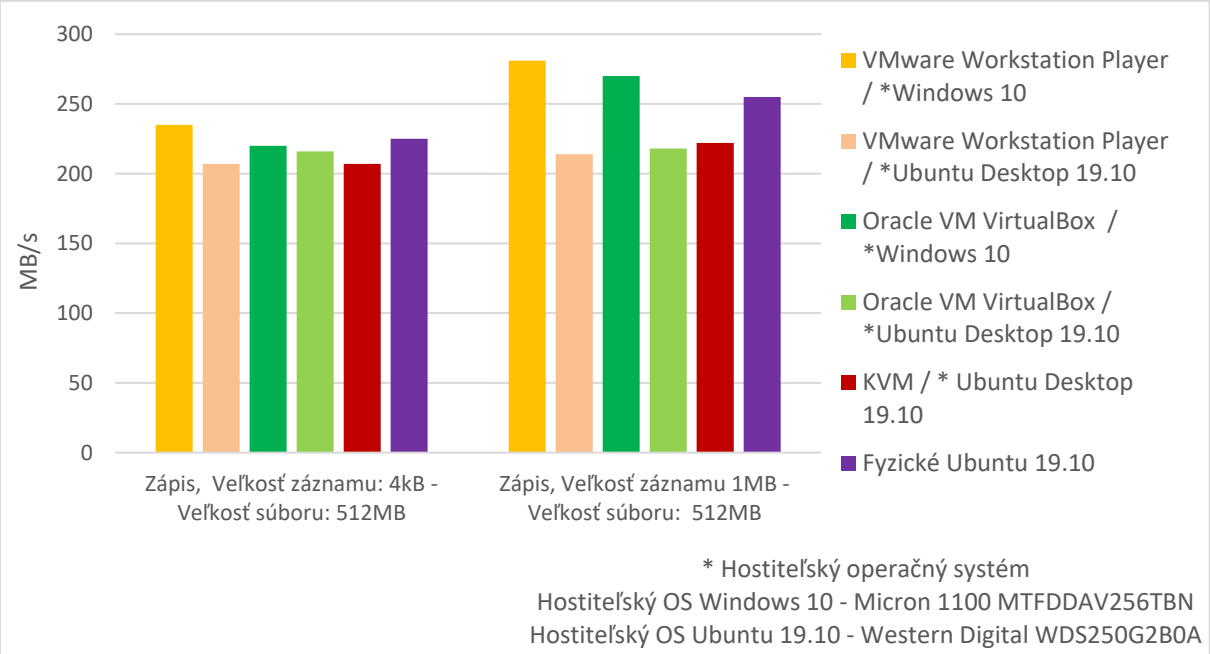
9.3.1 IOzone

Prvý test bol zameraný na čítanie. Ako môžeme pozorovať na Grafe 23, platformy VMware Workstation Player a Oracle VM VirtualBox s hostiteľským OS Windows 10 a rovnakým použitým SSD zaznamenali takmer identické výsledky. V prípade fyzickej inštalácie a ostatných platforiem využívajúcich Ubuntu 19.10 môžeme pozorovať vyrovnané výsledky u VMware Workstation a KVM, prepad približne o 25% oproti fyzickej inštalácií zaznamenala platforma Oracle VM VirtualBox.



Graf 23: IOzone Čítanie - Platformy pre OS Linux a Windows

Výsledky druhého testu so zameraním na zápis môžeme pozorovať na Grafe 24, medzi dvomi platformami s hostiteľským OS Windows si mierne lepšie viedol VMware Workstation Player v porovnaní s Oracle VM VirtualBox. U platforiem s hostiteľským OS Ubuntu 19.10 môžeme pozorovať veľmi podobné výsledky.



Graf 24: IOzone - Zápis - Platformy pre OS Linux a Windows

9.3.2 **Zhrnutie**

Výsledky jednotlivých platforiem sa nijak zvlášť nelíšili, jediný horší výsledok v teste zápisu zaznamenala platforma Oracle VM VirtualBox pri hostiteľskom OS Ubuntu 19.10. V tomto prípade boli k testom využité SSD disky, takže fragmentácia nehrala žiadnu rolu. Obidva použité SSD disky mám na svojom súkromnom počítači zaplnené skoro na maximum kapacity, čo by v prípade starších SSD spôsobilo problémy s výkonom, moje SSD ale využívajú vyrovnávaciu pamäť „DRAM", ktorá tento problém eliminuje.

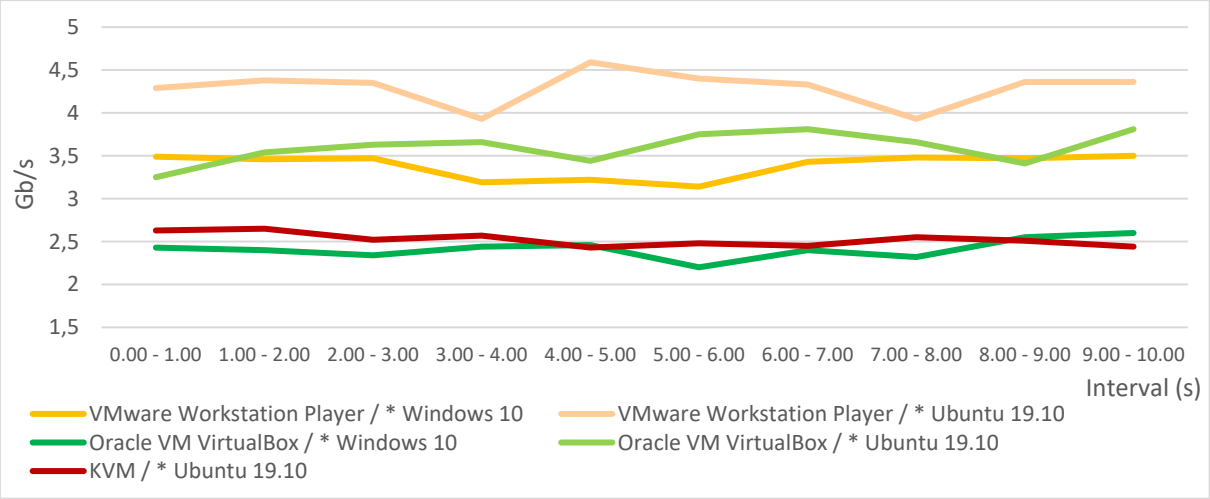
9.4 Test virtuálnej siete

Pre testovanie virtuálnej siete som zvolil nástroj iPerf3 a identickú konfiguráciu, ako v prípade serverovej virtualizácie. Tentokrát som sa zameral len na maximálnu prenosovú rýchlosť medzi virtuálnymi strojmi v rámci platformy, samozrejme pre obidva protokoly TCP a UDP. K testom som využil vždy dva virtuálne stroje v rámci hostiteľa, kde jeden slúžil ako server a druhý ako klient.

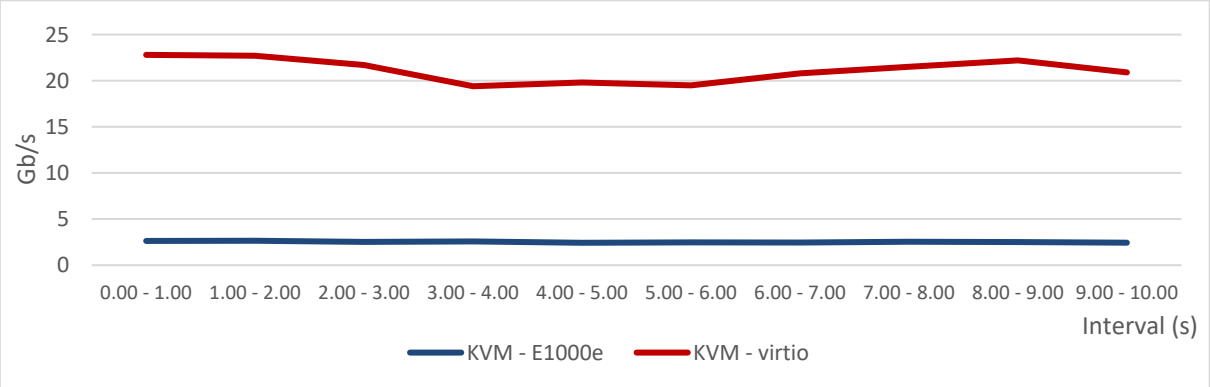
Každý test bol spustený päťkrát, výsledná a použitá hodnota bol aritmetický priemer nameraných výsledkov.

9.4.1 iPerf3 - TCP

Na Grafe 25 môžeme pozorovať, že najlepšie výsledky podávala platforma VMware Workstation Player s hostiteľským OS Ubuntu 19.10 s priemernou prenosovu rýchlosťou 4,2Gb/s. Platforma KVM podávala prenosové rýchlosti v priemere 2,5Gb/s, tu by som ale chcel upozorniť na zmenu typu virtuálneho sieťového adaptéru. Pri serverovej virtualizácií som používal predvolené moderné rozhranie virtio, Obrázok 43. Tentokrát som ale zmenil typ z virtio na E1000e, jedná sa o emulovanú verziu sieťového adaptéru Intel 82574. Porovnanie a dopad tejto zmeny môžeme pozorovať na Grafe 26, kde bola nameraná rýchlosť pri použití virtio mnohonásobne vyššia, ako pri použití E1000e.



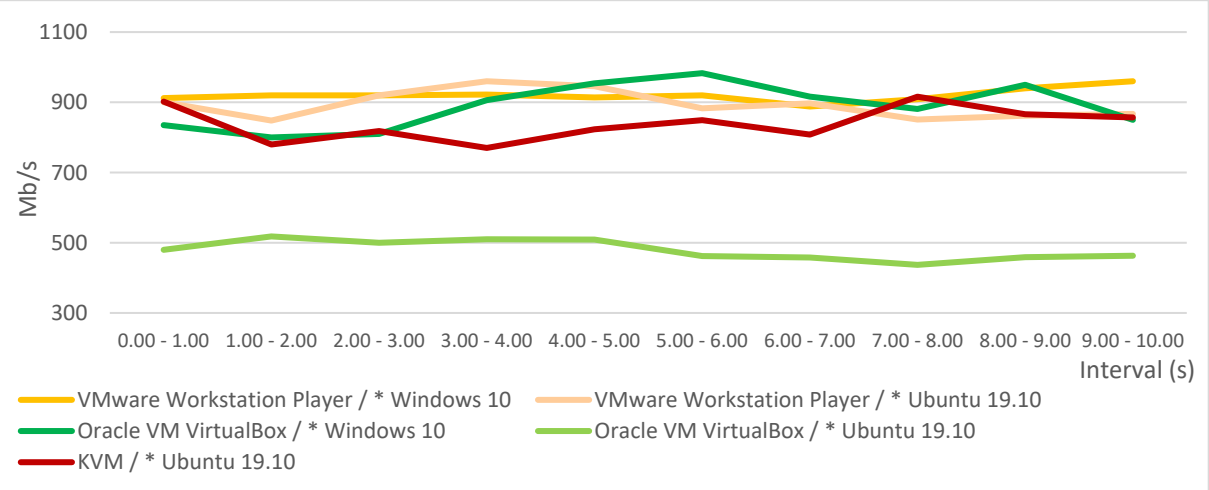
Graf 25: iPerf3 - TCP - Platformy pre OS Linux a Windows



Graf 26: iPerf3 TCP - KVM, porovnanie virtio a E1000e

9.4.2 **iPerf3 - UDP**

V test UDP môžeme pozorovať, že jednotlivé platformy podávali až na Oracle VM VirtualBox s hostiteľským OS Ubuntu 19.10 vyrovnané výsledky. V doplnkovej Tabuľke 7 môžeme pozorovať minimálnu strátovosť datagramov a nízky Jitter u každej z platforiem.



Graf 27: iPerf3 - UDP - Platformy pre OS Linux a Windows

Tabuľka 7: Doplnok ku Grafu 27 - Jitter a strátovosť datagramov

	Jitter (ms)	Stratený / Celkový počet datagramov
VMware Workstation Player / Windows 10	0,014	1351 /141790 (0,95%)
VMware Workstation Player / Ubuntu 19.10	0,016	474 / 136300 (0,35%)
Oracle VM Virtualbox / Windows 10	0,028	2253 / 133800 (1,68%)
Oracle VM Virtualbox / Ubuntu 19.10	0,042	133 / 72050 (0,18%)
KVM / Ubuntu 19.10	0,089	183 / 127240 (0,14%)

9.4.3 **Zhrnutie**

Zaujímavé výsledky sme mohli pozorovať hlavne pri teste protokolu TCP, kde ma prekvapil vplyv hostiteľského systému na prenosovú rýchlosť. Platformy VMware Workstation Player a Oracle VM VirtualBox s hostiteľským Ubuntu 19.10 podávali lepšie výsledky, ako pri hostiteľskom OS Windows 10. Taktiež sme mohli pozorovať, aký veľký vplyv môže mať voľba sieťového rozhrania pri platforme KVM, kde paravirtualizované rozhranie virtio podávalo mnohonásobné lepšie výsledky ako emulovaná verzia sieťového adaptéru Intel 82574.

10 Subjektívne porovnanie nástrojov využitých pre správu hostiteľa a virtuálnych strojov

Kapitola sa zaoberá subjektívnym porovnaním nástrojov pre správu hostiteľa a virtuálnych strojov, ktoré boli v práci využité.

10.1 Serverové virtualizačné platformy

V teoretickej časti boli popísané oficiálne, najpoužívanejšie alebo alternatívne nástroje pre správu hostiteľa a jeho hostí. Ja som využíval výhradne oficiálne, prípadne najznámejšie, ktoré sú pre danú platformu dostupné.

10.1.1 VMware ESXi - ESXi Embedded Host Client

Moderný klient, ktorý je súčasťou samostatného hypervizora VMware ESXi. Medzi jeho najvačšie výhody patrí z môjho pohľadu to, že sa jedná o webový klient postavený na HTML5 a je teda kompatibilný s každým moderným webovým prehliadačom, z čoho vyplýva aj výborná kompatabilita naprieč rôznymi OS. Dokonca som nemal problém so základnou správou prostredníctvom prehliadača Google Chrome nainštalovanom na mobilnom telefóne s OS Android. Samotný klient je veľmi intuitívny, či už hovoríme o prehľadnom sprievodcovi pri vytváraní virtuálneho stroja, vytváraní virtuálnych sietí alebo konfigurácií úložiska a následnom nahrávaní obrazov jednotlivých OS. Dostupné je aj prehľadné monitorovanie systémových prostriedkov hostiteľa alebo jeho hosťa, ale to len v rozmedzí jednej hodiny. Práve v ombedzenom monitorovaní systémových prostriedkov vidím jedinú nevýhodu tohoto inak výborného klienta.

10.1.2 Citrix Hypervisor - Citrix XenCenter

Voľne dostupná aplikácia Citrix XenCenter je na rozdiel od ESXi Embedded Host klienta štandardná aplikácia, ktorú si musí administrátor nainštalovať na svoje zariadenie odkiaľ spravuje infraštruktúru. Aplikácia je dostupná len pre OS Windows, mne tento fakt počas používania neprekážal, ale administrátorom, ktorí sú zvyknutí na iné OS môže toto obmedzenie skomplikovať prácu. Riešením môže byť aplikácia tretrej strany XenOrchestra alebo virtuálny stroj s OS Windows. Keď sa vrátime ku samotnému prostrediu aplikácie XenCenter, tak to na mňa pôsobilo síce mierne zastaraným dojmom, ale na druhú stranu bolo intuitívne a prehľadné. Za výborné považujem možnosti monitorovania systémových prostriedkov v rozmedzí 10 minút až 1 roka. Naopak, veľké negatívum vidím v zložitom a neprehľadnom vytváraní lokálneho úložiska pre obrazy jednotlivých OS prostredníctvom príkazu, kapitola 5.3.2.

10.1.3 Microsoft Hyper-V - Hyper-V Manager

Podobne, ako Citrix XenCenter je Hyper-V Manager štandardná aplikácia, ktorá je dostupná len pre operačný systém Windows. Prostredie Hyper-V Managera je veľmi podobné vstavaným aplikáciam desktopového OS Windows 10, ako „Správca diskov" alebo „Správca zariadení". Osobne mi tento typ prostredia nikdy nevyhovoval a ani pri Hyper-V Managerovi tomu nebolo inak. Najväčšie problémy som zaznamenal s prehľadnosťou prostredia pri konfigurácií virtuálnej siete. Jednotlivé prvky sú z môjho pohľadu neprehľadne a chaoticky rozdelené, prípadne zbytočne skryté v nastaveniach virtuálneho stroja.

Za veľkú nevýhodu považujem aj takmer nulové možnosti monitorovania systémových prostriedkov. Naopak, ako pozitívum považujem prehľadného sprievodcu vytvorením virtuálneho stroja alebo jednoduchú konfiguráciu správy dynamickej operačnej pamäte.

10.1.4 KVM/QEMU - virt-manager

Aplikácia virt-manager dostupná len pre OS Linux ponúka jednoduché používateľské rozhranie v ktorom sa mi počas práce ľahko orientovalo. Tak, ako v prípade Citrix XenCenter a Hyper-V Manager je kompatabilita len s jedným OS opäť mierne obmedzenie, riešenie spočíva napríklad vo virtuálnom stroji s OS Linux alebo v prípade dostupnosti aj vo vzdialenej ploche prostredníctvom protokolu VNC. Sprievodca vytvorením virtuálneho stroja mi prišiel oproti predchádzajúcim platformám viac priamočiary a aj jednoduchší, čo hodnotím veľmi kladne. Na to, že aplikácia môže pôsobiť jednoduchým dojmom, tak ponúka široké spektrum nastavení, ktoré z môjho pohľadu uspokoja aj pokročilého administrátora. Jedinú nevýhodu vidím podobne, ako pri Hyper-V Manager v monitorovaní systémovych prostriedkov, ktoré sa tu nachádza v úplne základnej forme.

10.1.5 LXD - klient lxc

LXD bola jediná platforma pri ktorej som využíval ku správe kompletne príkazový riadok bez grafického používateľského rozhrania. V prípade LXD sa jedná o lxc príkazy, ktoré nie sú vôbec komplexné a ľahko sa s nimi pracuje. K jednoduchosti prispieva aj výborne spracovaná dokumentácia, vďaka ktorej som sa rýchlo naučil spravovať, vytvárat a upravovať LXD kontajnery. Takýto klient pracujúci v príkazovom riadku je aj maximálne flexibilný naprieč OS, kde oprávnenému administrátorovi stačí ku správe pripojenie k hostiteľovi prostredníctvom SSH. Druhou možnosťou, ktorú som už spomenul v teoretickej časti je dostupnosť samostatného lxc klienta pre OS Linux, Windows a macOS. Celkovo, voči lxc klientovi nemám takmer žiadne výhrady, jedinou výhradou sú opäť obmedzené možnosti monitorovania systémových prostriedkov, kde sme obmedzení len na základné informácie o kontajneri prostredníctvom príkazu lxc info.

10.1.6 Zhrnutie

Počas relatívne dlhej doby za ktorú som mal možnosť pracovať s týmito piatimi klientami a z čisto subjektívneho hľadiska hodnotím takmer vo všetkých smeroch ako najlepší ESXi Embedded Host Client. To z dôvodu jeho moderného, prehľadného a logicky usporiadaného prostredia, ktoré ku svojej funkcií potrebuje len moderný webový prehliadač a tým je schopné bezproblémovo pracovať na širokej škále operačných systémov. Osobne sa mi výborne pracovalo aj s klientom lxc, ktorého syntax nie je zložitá a základné príkazy sú realtívne ľahko zapamätateľné.

10.2 Virtualizačné platformy pre OS Windows a Linux

Platformy VMware Workstation Player a Oracle VM VirtualBox zastupujúce hypervizor Typu 2 sú kompletné produkty, ktoré zahŕňajú vstavaný klient a iné, alternatívne možnosti ani nie sú dostupné. Nástroj virt-manager v tejto kapitole nie je zahrnutý, to z dôvodu, že skúsenosti s používaním boli identické so serverovou virtualizáciou.

10.2.1 VMware Workstation Player

Vstavaný klient VMware Workstation Player je už pohľadu cielený na jednoduchosť. Oproti klientom pri serverovej virtualizácií ponúka samozrejme ďaleko menšie možnosti nastavení. Môj názor je, že ponúkané možnosti budú cieľovej skupine používateľov vo väčšine prípadov postačovať. Osobne ma zaujal sprievodca inštaláciou virtuálneho stroja a jeho funkcia „Easy Install", ktorá pri hosťoch s OS Windows alebo vybraných distribúciach OS Linux ešte zjednoduší proces vytvárania a inštalácie.

10.2.2 Oracle VM VirtualBox

Mierne odlišným dojmom na mňa pôsobilo prostredie klienta Oracle VM VirtualBox, ktoré mi ani zďaleka neprišlo také prehladné, ako pri VMware Workstation Player. Osobne mi najviac vadil sprievodca vytvorením virtuálneho stroja, ktorý je veľmi strohý a väčšinu nastavení, vrátane cesty k obrazu OS je nutné po vytvorení nakonfigurovať priamo v nastaveniach. Na druhú stranu, nastavenia virtuálneho stroja ponúkajú širokú škálu možností a nastavení. Súčasťou klienta je aj VBoxManage, ktorý umožňuje základnú, ale aj pokročilú správu virtuálnych strojov priamo z príkazového riadku.

10.2.3 Zhrnutie

Každý z klientov na mňa pôsobil úplne diametrálne. Z môjho pohladu bolo hlavným cieľom pri vývoji klienta VMware Workstation Player jednoduchosť a zameranie skôr na menej skúsených používateľov. Opak som pociťoval pri používaní Oracle VM VirtualBox, kde pokročilé nastavenia smerujú skôr k skúsenejším používateľom.

Záver

V tejto diplomovej práci som sa snažil čitateľovi čo najviac priblížiť, predstaviť a porovnať moderné virtualizačné platformy VMware ESXi, Citrix Hypervisor, Microsoft Hyper-V, KVM, LXD, VMware Workstation Player a Oracle VM VirtualBox. Zameral som sa nie len na platformy pre operačné systémy Linux a Windows, ale hlavne na platformy z oblasti serverovej virtualizácie, ktoré sú v dnešnej dobe neoddeliteľnou súčasťou informačných technológií.

Úvod teoretickej časti bol venovaný popisu serverovej virtualizácie a jednotlivým virtualizačným technikám, ako plná virtualizácia, paravirtualizácia alebo virtualizácia na úrovni OS. Nasledoval popis vybraných serverových platforiem, ich funkcie, vlastnosti, architektúra, stručná história, rozdelenie z pohľadu produktovej rady alebo možnosti adminstrácie. Súčasťou teoretickej časti je aj popis možností využitia grafických kariet pre virtualizáciu.

Praktická časť bola venovaná inštalácií, konfigurácií a porovnaniu výkonu jednotlivých virtualizačných platforiem. Pri každej z platforiem sa nachádza popis tvorby virtuálneho stroja, možnosti tvorby virtuálnej siete a celková konfigurácia vybraných platforiem pred testovaním. Samotné testovanie zahŕňalo podrobný popis metodiky testovania, voľbu porovnávacích nástrojov, použité hardvérové a softvérové prostriedky, jednotlivé testy a krátke zhrnutie.

Po prečítaní práce si čitateľ môže položiť jednoduchú otázku, ktorá z platforiem z oboch oblastí virtualizácie je teda najlepšia. Odpoveď na túto otázku je naozaj zložitá a záleží od veľkého množstva rôznych faktorov a celkového uhla pohľadu. Keď sa na to pozrieme z čisto výkonového hľadiska, tak tu je voľba na čitateľovi a jeho úsudku na základe prečítania tejto práce.

Z celkového hľadiska začneme s VMware ESXi, ktorý je dostupný zdarma a počas testovania dosahoval stabilných výkonových výsledkov. Ponúka ideálne riešenie pre malé firmy, testovanie alebo vzdelávacie účely. Za jeho obrovskú výhodu považujem moderné a intuitívne webové používateľské rozhranie. Jediné väčšie mínus môžeme považovať maximum 8 vCPU pre každý vituálny stroj. Osobne si ale myslím, že je to naozaj malá daň za to, čo všetko ponúka.

Citrix Hypervisor Express Edition sa mi hodnotí relatívne obtiažne, z pohľadu výkonu podával spoločne s Microsoft Hyper-V menej presvedčivé výsledky, taktiež túto platformu podľa mňa zbytočne zráža relatívne zastaraný, aj keď intuitívny klient, ktorý je naviac dostupný len pre OS Windows. Môže nájsť podobné využitie, ako VMware ESXi v malých firmách alebo za účelom vzdelávania.

Nasleduje platforma Microsoft Server Hyper-V. Menej presvedčivé výsledky a z môjho pohľadu aj neprehľadný a chaotický klient Hyper-V Manager (ktorý by ale v budúcnosti mohol byť nahradený moderným webovým prostredím Windows Admin Center) vo mne zanechali zmiešané pocity. Na druhú stranu je potrebné poznamenať fakt, že hostia využívali vždy OS Linux. V prípade použitia hostí s OS Windows môže byť situácia v testoch odlišná. Microsoft Hyper-V si určite nájde miesto hlavne v prostrediach, kde sa vyžaduje nasadenie virtuálnych strojov práve s OS Windows.

Ďalšou platformou je KVM, ktorá je neoddeliteľnou súčasťou Linux jadra. Stabilné výsledky, až prekvapujúco jednoduchá inštalácia a počiatočná konfigurácia robia z tejto platformy výbornú alternatívu k už spomenutým riešeniam. Vďaka integrácií do Linux jadra sa výborne hodí pre virtuálne stroje s OS Linux.

Svojou flexibilitou nájde uplatnenie nie len v podnikovej sfére, kde má zastúpenie napríklad v podobe Red Hat Virtualization, ale aj pri použití v klasických stolných počítačoch s rôznymi distribúciami OS Linux.

Posledným, ale úplne odlišným zástupcom serverovej virtualizácie je kontajnerová platforma LXD. Je práve tou, ktorá ma zaujala najviac. Stabilné výkonnostné výsledky, jednoduchá inštalácia, konfigurácia a správa robia z tejto platformy v určitých oblastiach využitia lepšiu alternatívu, ako napríklad KVM. Najlepším príkladom je podľa mňa vzdelávanie a testovanie. Z vlastnej skúsenosti môžem uviesť predmety Praktikum komunikačných sietí I a II, ktoré sú vyučované na Katedre telekomunikačnej techniky. Počas týchto predmetov sme mali k dispozícii KVM virtuálne stroje slúžiace k realizácií rôznych projektov alebo k testovaniu. Práve pri takýchto situáciach si myslím, že LXD môže smelo nahradiť KVM. Pre niektorých môže byť mínusom LXD kontajnerov logická a vyplývajúca podpora kontajnerov len s OS Linux. LXD rýchlo napreduje, dôkazom toho je aj veľmi aktívny vývoj zo strany spoločnosti Canonical a od verzie LXD 4.0 LTS vydanej 3.4.2020 bola dokonca pridaná podpora pre plnohodnotné virtuálne stroje.

Posledné platformy sú zástupcovia virtualizačných platforiem pre OS Windows a Linux, jedná sa o VMware Workstation Player a Oracle VM VirtualBox. Nakoľko sa jedná o platformy pre klasické stolné počítače určené hlavne pre domáce použitie, tak tu by som osobne nebral výkon ako hlavný údaj na ktorom záleží výber. Osobne si myslím, že v drvivej väčšine sú tieto platformy používané ako testovacie prostredia rôznych OS alebo aplikácií a tak voľba zaleží hlavne na používateľovi a jeho preferenciách. Pokiaľ chceme jednoduchosť, voľba je VMware Workstation Player, pre skúsenejších používateľov bude zasa vhodnejší komplexnejší Oracle VM VirtualBox.

Z pohľadu hlbšieho porovnania sa samozrejme ponúkajú aj komplexnejšie testy pri rôznom zaťažení serveru, počte spustených virtuálnych strojov, rôznych scenároch konfigurácie siete alebo služieb. Pri takomto porovnaní si ale myslím, že by bolo vhodnejšie porovnávať ideálne dve, najviac tri virtualizačné platformy.

Pevne verím, že táto diplomová práca prinesie každému čitateľovi užitočné informácie z tejto oblasti a prípadne pomôže pri konfigurácií alebo dokonca výbere virtualizačnej platformy.

Použitá literatúra

[1] Čo je virtualizácia? [online]. [cit.2020-01-09].

Dostupné z: http://www.v-portal.sk/2011/01/co-je-to-virtualizacia/

[2] Virtualizácia [online]. [cit.2020-01-09].

Dostupné z: https://www.digitalmag.sk/virtualizacia/

[3] Type 1 and Type 2 Hypervisors: Whats Makes The Different [online]. [cit.2020-01-09].

Dostupné z: https://medium.com/teamresellerclub/type-1-and-type-2-hypervisors-what-makes-them-different-6a1755d6ae2c

[4] Types of Server Virtualization In Computer Network [online]. [cit.2020-01-09].

Dostupné z: https://www.geeksforgeeks.org/types-of-server-virtualization-in-computer-network/

[5] Techniky virtualizace počítačů (2) [online]. [cit.2020-01-09].

Dostupné z: http://webserver.ics.muni.cz/bulletin/articles/545.html

[6] Para virtualization vs Full virtualization vs Hardware Virtualization [online]. [cit.2020-01-12].

Dostupné z: https://www.unixarena.com/2017/12/para-virtualization-full-virtualization-hardware-assisted-virtualization.html/

[7] Virtualizace [online]. [cit.2020-01-12].
Dostupné z: https://www.fi.muni.cz/~kas/pv090/referaty/2016-podzim/virt.html#podcast4

[8] VMware [online]. [cit.2020-01-12].
Dostupné z: https://searchvmware.techtarget.com/definition/VMware

[9] VMWARE vSPHERE COMPUTE VIRTUALIZATION [online]. [cit.2020-01-12].
Dostupné z: https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/products/vsphere/vmware-vsphere-vsom-pricing-whitepaper.pdf

[10] Free VMware ESXi: Restrictions and Limitations [online]. [cit.2020-01-15].
Dostupné z: https://www.nakivo.com/blog/free-vmware-esxi-restrictions-limitations/

[11] The Architecture of VMware ESXi [online]. [cit.2020-01-22].
Dostupné z: https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/ESXi_architecture.pdf

[12] What is VMware vCenter Server? [online]. [cit.2020-01-22].

Dostupné z: http://www.mustbegeek.com/what-is-vmware-vcenter-server/

[13] An Introductory Guide to vSphere HTML5 Client [online]. [cit.2020-01-23].
Dostupné z: https://www.altaro.com/vmware/vsphere-html5-client/

[14] ESXi Embedded Host Client [online]. [cit.2020-01-23].

Dostupné z: https://flings.vmware.com/esxi-embedded-host-client

[15] Everything you need to know about Citrix [online]. [cit.2020-01-23].

Dostupné z: https://www.itpro.co.uk/saas/28932/everything-you-need-to-know-about-citrix

[16] Licensing [online]. [cit.2020-01-23].

Dostupné z: https://docs.citrix.com/en-us/citrix-hypervisor/overview-licensing.html

[17] Technical overview [online]. [cit.2020-01-23].

Dostupné z: https://docs.citrix.com/en-us/citrix-hypervisor/technical-overview.html

[18] What is XenCenter? [online]. [cit.2020-01-23].
Dostupné z: https://subscription.packtpub.com/book/virtualization_and_cloud/9781849689823/1/ch01lvl1sec14/what-is-xencenter

[19] Managing XenServer with XenCenter and Xen Orchestra Web Interfaces - Part -7 [online]. [cit.2020-01-23].
Dostupné z: http://www.wtomasini.nl/wordpress/managing-xenserver-with-a-xencenter-and-xen-orchestra-web-interfaces-part-7/

[20] What is Hyper-V? [online]. [cit.2020-01-24].

Dostupné z: https://www.altaro.com/hyper-v/what-is-hyper-v/

[21] Standard and Datacenter Windows Server 2016 Versions: Overview [online]. [cit.2020-01-24].

Dostupné z: https://www.nakivo.com/blog/standard-and-datacenter-windows-server-2016-versions-overview/

[22] Windows Server 2019 Essentials bude posledný v rade. [online]. [cit.2020-01-24].

Dostupné z: https://www.pcrevue.sk/a/Windows-Server-2019-Essentials-bude-posledny-v-rade--Zakaznici-po-ukonceni-podpory-budu-musiet-prejst-na-nieco-ine

[23] Hypervisors [online]. [cit.2020-01-27].

Dostupné z: https://www.sciencedirect.com/topics/computer-science/hypervisors

[24] Appendix B: Hyper-V Architecture and Feature Overview [online]. [cit.2020-01-27].

Dostupné z: https://docs.microsoft.com/en-us/biztalk/technical-guides/appendix-b-hyper-v-architecture-and-feature-overview

[25] What IS Hyper-V Manager and How Does It Work? [online]. [cit.2020-01-27].
Dostupné z: https://www.nakivo.com/blog/what-is-hyper-v-manager-and-how-does-it-work/

[26] Siron Eric. HOW TO GET THE MOST OUT OF WINDOWS ADMIN CENTER [eKniha]. [online]. [cit.2020-01-28].

Dostupné z: https://www.altaro.com/ebook/windows-admin-center.php

[27] What is KVM? [online]. [cit.2020-01-28].

Dostupné z: https://www.redhat.com/en/topics/virtualization/what-is-KVM

[28] KVM FAQ [online]. [cit.2020-01-28].

Dostupné z: http://www.linux-kvm.org/page/FAQ#FAQ

[29] What Is the Difference between QEMU and KVM? [online]. [cit.2020-01-28]. Dostupné z: https://www.fir3net.com/UNIX/Linux/what-is-the-difference-between-qemu-and-kvm.html

[30] libvirt [online]. [cit.2020-01-28].

Dostupné z: https://wiki.archlinux.org/index.php/Libvirt

[31] Manage virtual machines with virt-manager [online]. [cit.2020-01-28].

Dostupné z: https://virt-manager.org/

[32] KVM/Virsh [online]. [cit.2020-01-28].

Dostupné z: https://help.ubuntu.com/community/KVM/Virsh

[33] kimchi-project [online]. [cit.2020-01-29].

Dostupné z: https://github.com/kimchi-project/kimchi

[34] What`s a Linux container? [online]. [cit.2020-01-29].

Dostupné z: https://www.redhat.com/en/topics/containers/whats-a-linux-container

[35] What`s the Diff: VMs vs Containers [online]. [cit.2020-01-29].

Dostupné z: https://www.backblaze.com/blog/vm-vs-containers/

[36] What`s LXC? [online]. [cit.2020-01-29].

Dostupné z: https://linuxcontainers.org/lxc/introduction/#LXC

[37] What is the difference between a process, a container, and a VM? [online]. [cit.2020-01-29].

Dostupné z: https://medium.com/@jessgreb01/what-is-the-difference-between-a-process-a-container-and-a-vm-f36ba0f8a8f7

[38] LXC - Security [online]. [cit.2020-01-29].

Dostupné z: https://linuxcontainers.org/lxc/security/

[39] LXC and LXD: Explaining Linux Containers [online]. [cit.2020-01-29].

Dostupné z: https://www.sumologic.com/blog/lxc-lxd-linux-containers/

[40] What`s LXD? [online]. [cit.2020-01-29].

Dostupné z: https://linuxcontainers.org/lxd/introduction/

[41] Getting started with LXD [online]. [cit.2020-01-30].
Dostupné z: https://events19.linuxfoundation.org/wp-content/uploads/2017/11/Getting-Started-with-LXD-and-System-Containers-St%C3%A9phane-Graber-_-Christian-Brauner-Canonical-Ltd.-.pdf

[42] LXC vs Docker - What`s the best for your website? [online]. [cit.2020-01-30].

Dostupné z: https://bobcares.com/blog/lxc-vs-docker/

[43] LXD - How can I run docker inside a LXD container? [online]. [cit.2020-01-31].

Dostupné z: https://lxd.readthedocs.io/en/latest/#support-and-discussions

[44] Lxdui [online]. [cit.2020-01-31].

Dostupné z: https://github.com/AdaptiveScale/lxdui

[45] What Is a Virtual GPU? [online]. [cit.2020-02-04].

Dostupné z: https://blogs.nvidia.com/blog/2018/06/11/what-is-a-virtual-gpu/

[46] NVIDIA VIRTUAL GPU TECHNOLOGY [online]. [cit.2020-02-04].

Dostupné z: https://www.nvidia.com/en-us/data-center/virtual-gpu-technology/

[47] NVIDIA VIRTUAL GPU PACKAGING, PRICING AND LICENSING [online]. [cit.2020-02-04].
Dostupné z: https://images.nvidia.com/content/grid/pdf/Virtual-GPU-Packaging-and-Licensing-Guide.pdf

[48] Dust Free! NVIDIA GRID and a GPU Deep Dive: Guest Blog Post by Richar d Hoffman [online]. [cit.2020-02-04].
Dostupné z: http://blog.itvce.com/2016/03/22/dust-free-nvidia-grid-and-a-gpu-deep-dive-guest-blog-post-by-richard-hoffman/

[49] VIRTUAL GPU SOFTWARE DOCUMENTATION [online]. [cit.2020-02-04].

Dostupné z: https://docs.nvidia.com/grid/latest/grid-vgpu-user-guide/index.html

[50] AMD MxGPU aims to give GRID a run for its money [online].[cit.2020-02-06].

Dostupné z: https://www.brianmadden.com/opinion/AMD-MxGPU-aims-to-give-GRID-a-run-for-its-money

[51] SOLUTION BRIEF AMD MULTIUSER GPU [online]. [cit.2020-02-06].
Dostupné z: https://www.amd.com/system/files/documents/amd-mxgpu-solution-brief.pdf

[52] Bringing New Use Cases and Workloads to the Cloud with Intel Graphics Virtualization Technology (Intel GVT-g) [online]. [cit.2020-02-06].

Dostupné z: https://01.org/sites/default/files/downloads/igvt-g/gvtflyer.pdf

[53] Intel Graphics Virtualization Update [online]. [cit.2020-02-06].

Dostupné z: https://software.intel.com/en-us/blogs/2014/05/02/intel-graphics-virtualization-update

[54] Rufus - Create bootable USB drives easy way. Dostupné z: https://rufus.ie/

[55] Whats Is VMware vSwitch? [online]. [cit.2020-02-19].

Dostupné z: https://www.nakivo.com/blog/what-is-vmware-vswitch/

[56] Network Adapters Type [online]. [cit.2020-02-27].

Dostupné z: https://docs.vmware.com/en/VMware-vSphere/6.5/com.vmware.vsphere.vm_admin.doc/GUID-AF9E24A8-2CFA-447B-AC83-35D563119667.html

[57] Networking [online]. [cit.2020-02-28].

Dostupné z: https://docs.citrix.com/en-us/citrix-hypervisor/networking.html

[58] Hyper-V should be the only enabled role [online]. [cit.2020-03-05].

Dostupné z: https://docs.microsoft.com/en-us/windows-server/virtualization/hyper-v/best-practices-analyzer/hyper-v-should-be-the-only-enabled-role

[59] Should I create a generation 1 or 2 virtual machine in Hyper-V? [online]. [cit.2020-03-05].

Dostupné z: https://docs.microsoft.com/en-us/windows-server/virtualization/hyper-v/plan/should-i-create-a-generation-1-or-2-virtual-machine-in-hyper-v

[60] What is the Hyper-V Virtual Switch and How Does it Work? [online]. [cit.2020-03-05].

Dostupné z: https://www.altaro.com/hyper-v/the-hyper-v-virtual-switch-explained-part-1/

[61] KVM/Networking [online]. [cit.2020-03-05].

Dostupné z: https://help.ubuntu.com/community/KVM/Networking

[62] Virtio [online]. [cit.2020-03-09].

Dostupné z: https://wiki.libvirt.org/page/Virtio

[63] Linux Containers - Image Server

Dostupné z: https://us.images.linuxcontainers.org/

[64] Network management with LXD (2.3+) [online]. [cit.2020-03-09].

Dostupné z: https://stgraber.org/2016/10/27/network-management-with-lxd-2-3/

[65] LXD 3.19 has been released. [online].

Dostupné z: https://discuss.linuxcontainers.org/t/lxd-3-19-has-been-released/6529

[66] Understanding Common Networking Configurations [online]. [cit.2020-03-10].

Dostupné z: https://pubs.vmware.com/workstation11/index.jsp?topic=%2Fcom.vmware.ws.using.doc%2FGUID-D9B0A52D-38A2-45D7-A9EB-987ACE77F93C.html

[67] Network Configuration in VirtualBox [online]. [cit.2020-03-10].

Dostupné z: https://www.thomaskrenn.com/en/wiki/Network_Configuration_in_VirtualBox

[68] Geekbench Browser - Processor Benchmarks [online]. [cit.2020-03-20]. Dostupné z: https://browser.geekbench.com/processor-benchmarks

[69] Sysbench [online]. [cit.2020-03-20].

Dostupné z: http://manpages.ubuntu.com/manpages/trusty/man1/sysbench.1.html

[70] Timed Linux Kernel Compilation.

Dostupné z: https://openbenchmarking.org/test/pts/build-linux-kernel

[71] Phoronix Test Suite. Dostupné z: https://www.phoronix-test-suite.com/

[72] RAMspeedSMP. Dostupné z: https://openbenchmarking.org/test/pts/ramspeed-1.4.2

[73] IOzone. Dostupné z: https://openbenchmarking.org/test/pts/iozone-1.9.5ň

[74] Unpacking The Linux Kernel. Dostupné z: https://openbenchmarking.org/test/pts/unpack-linux

[75] Allocate Memory Resources to a Virtual Machine in the VMware Host Client. [online]. [cit.2020-04-13].

Dostupné z: https://docs.vmware.com/en/VMwarevSphere/6.5/com.vmware.vsphere.vm_admin.doc/GUID-49D7217C-DB6C-41A6-86B3-7AFEB8BF575F.html

[76] Dynamic Memory Control (DMC). [online]. [cit.2020-04-13].

Dostupné z: https://docs.citrix.com/en-us/xencenter/7-1/dmc-about.html

[77] Managing VM RAM better with Hyper-V dynamic memory. [online]. [cit.2020-04-13].

Dostupné z: https://www.veeam.com/blog/hyper-v-dynamic-memory-managing-vm-ram.html

[78] Is dynamic memory management for guests supported? [online]. [cit.2020-04-13]. Dostupné z: https://www.linux-kvm.org/page/FAQ#Is_dynamic_memory_management_for_guests_supported.3F

[79] iPerf3 - The ultimate speed test tool for TCP, UDP and SCTP. [online]. [cit.2020-04-13]. Dostupné z: https://iperf.fr/iperf-download.php

[80] PING. Dostupné z: http://manpages.ubuntu.com/manpages/cosmic/man8/ping.8.html

[81] NETDATA - Monitor everything in real-time. Dostupné z: https://www.netdata.cloud/

[82] wrk - a http benchmarking tool. Dostupné z: https://github.com/wg/wrk

[83] NGINX. Dostupné z: https://www.nginx.com/

Zoznam príloh

- Príloha A obsahuje kompletný výpis súboru dhcpd.conf.
- Príloha B obsahuje štatistiky testu z kapitoly 8.4.4 zachytené nástrojom NETDATA.